RiverWare

Technical Documentation Version 7.3

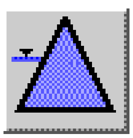
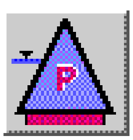
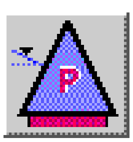
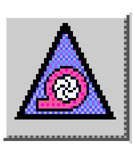
Objects

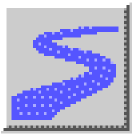
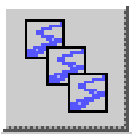
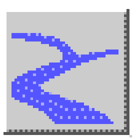
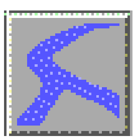
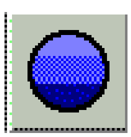
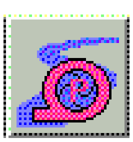
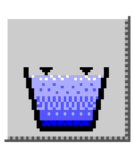
Center for Advanced Decision Support for Water and Environmental Systems (CADSWES)
UNIVERSITY OF COLORADO BOULDER

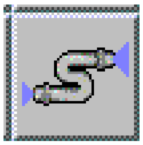
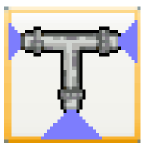
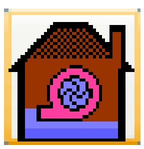
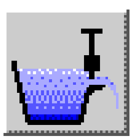
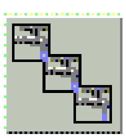
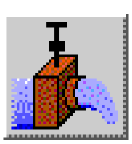
Objects
Table of Contents

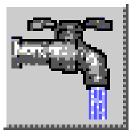
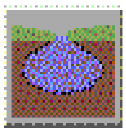
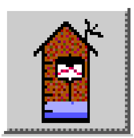
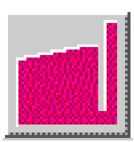
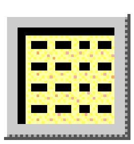
1. Object Types

The workspace contains the objects from which the model is constructed. Each object represents a physical feature of the river basin. Objects of the same type are customized by choosing methods and specifying data. There are two main classes of objects; simple objects and aggregate objects. A simple object represents a single physical feature of a river basin while an aggregate object represents multiple features of the same type. For example, an aggregate reach represents multiple reaches which may have different characteristics. The object types available in RiverWare are described on the following page:

ICON	OBJECT AND DESCRIPTION	LINK
	**Storage Reservoir** A reservoir with Release and spillways and no hydropower facilities. Storage is a function of Pool Elevation as defined by an Elevation-Volume Table.	Click HERE (Section 24) for more information
	**Level Power Reservoir** A reservoir with a hydropower plant (Turbine Release) and spillways. Storage is a function of Pool Elevation as defined by an Elevation-Volume Table.	Click HERE (Section 17) for more information
	**Slope Power Reservoir** A reservoir with hydropower facilities and spillways. Storage is a combination of level storage and wedge storage. Wedge storage is defined by a table which relates headwater elevation and Inflow to a water surface profile.	Click HERE (Section 23) for more information
	**Pumped Storage** A reservoir which has reversible pump-turbines. The turbines may generate or pump at each timestep. Storage is a function of Pool Elevation as defined by an Elevation-Volume Table.	Click HERE (Section 21) for more information

ICON	OBJECT AND DESCRIPTION	LINK
	**Reach** A river section which routes water using one of many possible routing algorithms. Reaches may lose water to a Diversion and gain water from Return Flow. Reaches can also have side inflows, gains, and/or losses.	Click HERE (Section 22) for more information
	**Aggregate Reach** An aggregate object which contains one or more Reach objects.	Click HERE (Section 4) for more information
	**Confluence** A flow junction with two Inflows and a single Outflow.	Click HERE (Section 8) for more information
	**Bifurcation** A flow junction with a single Inflow and two Outflows.	Click HERE (Section 5) for more information
	**Control Point** An object used to regulate upstream reservoirs so that channel capacity at the control point is not violated.	Click HERE (Section 9) for more information
	**Inline Power** An object used to model power production on a stretch of reach with no storage (run of river power production).	Click HERE (Section 15) for more information
	**Canal** A bi-directional conveyance channel which delivers water by gravity between two Reservoirs.	Click HERE (Section 6) for more information

ICON	OBJECT AND DESCRIPTION	LINK
	**Pipeline** An object that models flow in a pipeline between two objects.	Click HERE (Section 19) for more information
	**Pipe Junction** An object used with pressurized flow to split flows similar to a bifurcation or bring flows together similar to a confluence.	Click HERE (Section 18) for more information
	**Inline Pump** An object used to model a booster pump station. It controls solution direction, calculates added head and calculates the power consumed.	Click HERE (Section 16) for more information
	**Aggregate Distribution Canal** An aggregate object which serves to route Diversion Requests from a Water User upstream to a Diversion object. It also routes flow from the Diversion object down to the Water Users. The Aggregate Distribution Canal contains Distribution Canal element object. Their functionality is described HERE (Objects.pdf, Section 11.1).	Click HERE (Section 2) for more information
	**Aggregate Diversion Site** An aggregate object which contains zero or more Water Users. It diverts water from a Reach or a Reservoir. The Water User elements consume water and return excess flow to the system.	Click HERE (Section 3) for more information
	**Diversion Object** An object which diverts water from a Reservoir or Reach. The amount of water which may be diverted is based on water surface elevation, pumping parameters, or available water.	Click HERE (Section 12) for more information

ICON	OBJECT AND DESCRIPTION	LINK
	**Water User** An object that diverts water from a Reach or a Reservoir, consumes water, and then returns excess flow to the system.	Click HERE (Section 27) for more information
	**Ground Water Storage** An underground storage reservoir which receives Inflow from Water User Return Flow or Reach seepage and can return water to the system.	Click HERE (Section 14) for more information
	**Stream Gage** An object used to represent stream gage location. It shows the discharge data at a particular location in a model.	Click HERE (Section 25) for more information
	**Power Plant Diversion** An object which models the diversion and consumption of water by a Power Plant, often for cooling.The Power Plant Diversion is an aggregate object that contains Generator element objects. Their functionality is described HERE (Objects.pdf, Section 13.1).	Click HERE (Section 20) for more information
	**Thermal Object** An object which models the economics of the thermal power system and the thermal replacement value of the hydropower.	Click HERE (Section 26) for more information
	**Data Object** A container for custom slots and user defined data that is not appropriate on a simulation object.	Click HERE (Section 10) for more information

2. Agg Distribution Canal

The Aggregate Distribution Canal object contains Distribution Canal objects as its elements. The aggregated object may route the Diversion Requested from WaterUsers through a set of Distribution Canals upstream to a Diversion Object, Water User, or Agg Diversion Site. The AggDistributionCanal also routes flow through the Distribution Canals. Links are automatically created between the AggDistributionCanal and its elements. Links are also created and maintained between the Distribution Canal elements. A description of the linking structure is given in the Distribution Canal section.

General slots

(slots which always appear for this object)

☞ TOTAL INFLOW

Type: SeriesSlot

Units: FLOW

Description: flow entering the AggDistributionCanal object

Information: This slot can be set as an input, particularly if there are no objects upstream.

I/O: Optional; Required if not linked

Links: May be linked to the Outflow slot on a Diversion Object, the Diversion slot on any object, the Total Diversion slot on an AggDiversion Site, or the Total Outflow slot on an upstream AggDistributionCanal.

☞ TOTAL OUTFLOW

Type: SeriesSlot

Units: FLOW

Description: flow leaving the AggDistributionCanal object

Information: The values in this slot are set via propagation from the Outflow slot of the last Distribution Canal element in the AggDistributionCanal object.

I/O: Output only

Links: May be linked to the Total Inflow slot on a downstream AggDistributionCanal, the Inflow slot on any object, or the Return Flow slot on any object.

2.1 User Methods

2.1.1 Request Routing

Time Lag request routing is available within the Request Routing category. This category activates and deactivates the request routing calculations. If Time Lag request routing is selected, only Time Lag routing is available in the Flow Routing category on the Distribution Canal elements.

2.1.1.1 No Routing

This method does not route requests. No restriction is placed on the flow routing methods available on the Distribution Canal elements. There are no slots specifically associated with this method.

2.1.1.2 Time Lag

When the Time Lag request routing method is selected, all of the Distribution Canal elements are changed to the Time Lag routing method in the Flow Routing category.

This method routes the Delivery Requests upstream through all the Distribution Canal elements to the upstream end of the AggDistributionCanal. The method starts at the most downstream canal element and routes the Delivery Requests of that canal element and any Downstream Delivery Requests to the Routed Delivery Request slot at the top of that canal element. Once this has been accomplished, the values in the Routed Delivery Request slot are propagated along an automatic internal link to the DS Delivery Request slot of the upstream Canal element. When the most upstream canal element is reached, the Routed Delivery Request slot is propagated through another internal link to the Total Delivery Request slot on the AggDistributionCanal object.

Slots Specific to this Method

☞ Downstream Delivery Request

Type: SeriesSlot

Units: FLOW

Description: represents the delivery request from a downstream AggDistributionCanal

Information: The Downstream Delivery Request slot is automatically linked to the DS Delivery Request slot on the last Distribution Canal element on the AggDistributionCanal.

I/O: Optional; Set to zero if not input and not linked

Links: May be linked to the Total Delivery Request slot on a downstream AggDistributionCanal, the Diversion Requested slot on a WaterUser, or the Total Diversion Requested slot on an Agg Diversion Site.

☞ TOTAL DELIVERY REQUEST

Type: SeriesSlot

Units: FLOW

Description: sum of the Delivery Requests routed to the upstream end of the AggDistributionCanal

Information: The values in this slot are set via propagation from the Routed Delivery Request slot of the most upstream Distribution Canal element.

I/O: Output only

Links: May be linked to the Diversion Request slot on a Diversion object, the Total Diversion Requested slot on an Agg Diversion Site, or the Diversion Requested slot on a WaterUser.

The values in the Total Delivery Request slot are calculated in the following manner:

$$\text{Lag Integer} = \frac{\text{Lag Time}}{\text{Time Step}}$$

$$\text{total request} = \text{Delivery Request} + \text{DS Delivery Request}$$

The total request may also include evaporation and change in storage if those methods are active.

$$\text{Routed Delivery Request} = \frac{\text{total request}}{1 - \text{(Variable) Seepage Flow Fraction}}$$

Note: Lag Time, Delivery Request, Routed Delivery Request, (Variable) Seepage Flow Fraction, and DS Delivery Request are slots on the Distribution Canal element.

The Routed Delivery Request calculated in the equation above is then set for the timestep, t - Lag Integer, where t is the current timestep.

NOTE: If a method is selected in the Canal Storage category, the Routed Delivery Request is always zero when the canal is draining. In other words, no downstream or delivery requests can be routed upstream when the canal is draining.

The Routed Delivery Requested value is propagated across a link to the DS Delivery Request slot on the Canal element just upstream. The process above is then repeated until Routed Delivery Request is calculated on the most upstream Canal element. The value then propagates across a link to the Total Delivery Request slot on the AggDistributionCanal object.

This method executes at the beginning of the run. Therefore, it is necessary that all delivery requests on all elements are known at the beginning of the run.

2.2 Dispatch Methods

If No Routing is selected in the Request Routing category, this object does not dispatch.

2.2.1 routeRequests

This dispatch method is only executed when Time Lag request routing is selected in the Request Routing category.

Required Knowns

☞ **Downstream Delivery Request**

Required Unknowns

☞ **Total Inflow**

The AggDistributionCanal only dispatches on the first timestep. The dispatch method calls the function to route delivery requests upstream from the canal elements for all timesteps. This function is described in User Methods for the AggDistributionCanal object.

3. Agg Diversion Site

The Aggregate Diversion Site (Agg Diversion Site) object contains Water User objects as elements. The aggregate object allows many Water Users with separate diversions, return flows, and consumptive uses to be combined on the **RiverWare**™ Workspace to simplify model representation.

There are three possible "structures" defined on the Agg Diversion Site: the **Sequential Structure**, the **Lumped Structure**, and the **No Structure** (default). The linking structures define how the Water User objects behave relative to each other and their containing Agg Diversion Site. Links between the Agg Diversion Site object and the reach or reservoir object to which it is connected are made by the user. Links among the Water Users and between the Agg Diversion Site and the Water Users are done automatically according to the linking structure that the user has selected. The User Method section explains how to select the different linking structures.

3.1 User Methods

3.1.1 Link Structure

The Link Structure category, unlike most User Method Categories, is selected from the menu bar at the top of the Open Object dialog for the Agg Diversion Site. As described previously, there are three User Methods in the Link Structure category which define how the Water User objects behave relative to each other and their containing Agg Diversion Site object. Following is a description of each user method or "structure" and the associated slots and engineering calculations.

3.1.1.1 No Structure

The No Structure method is commonly referred to as the Stand-Alone or Default structure. This method treats the Agg Diversion Site object as a container for separate, stand-alone, Water Users. Each Water User is linked, by the user, directly to the reach or reservoir from which it is diverting water. However, each reach or reservoir Diversion slot or Available for Diversion slot should have only one link, meaning that a reach or reservoir may be linked to one and only one Water User for the No Structure method. This approach is good for highly diverted, highly regulated rivers where each individual diversion and return must be kept track of in the river reaches.

Since the Water Users are directly linked to a reach or reservoir, the Agg Diversion Site never dispatches. Therefore, there are no input slots. The figure below shows the linking structure.

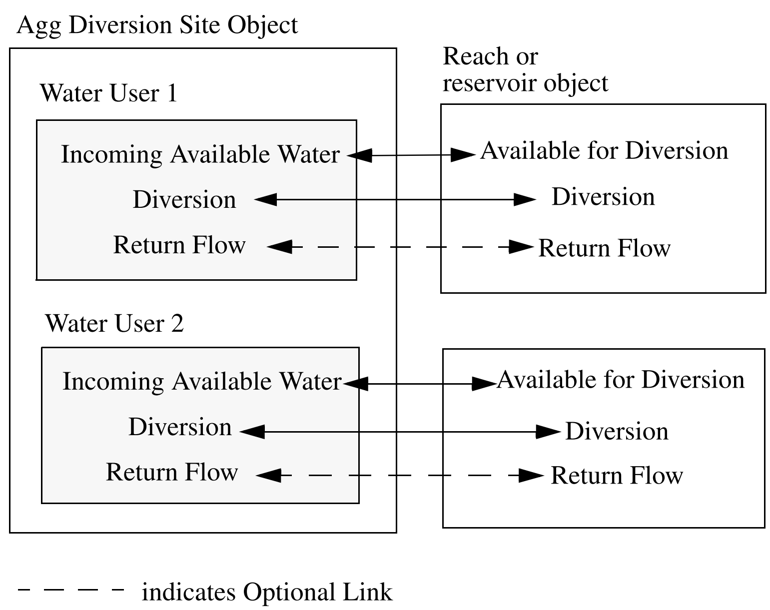


Figure: 1 Stand-Alone Links

Slots Specific to this Method

☞ Total Depletion Requested

Type: Multislot
Units: FLOW
Description: a multislot showing the Depletion Requested for each Water User as well as the sum of all requested depletions.
Information: Used for all linking structures.
I/O: Optional; Can be input when there are no Water User elements and it is not linked.
Links: Usually not linked

☞ Total Diversion Requested

Type: MultiSlot
Units: FLOW
Description: a multislot showing the Diversion Requested for each Water User as well as the sum of all requested diversions.
Information: Used for all linking structures.
I/O: Optional; Can be input when there are no Water User elements and it is not linked.

Links: May be linked to the Diversion Request slot on a Diversion Object, the Downstream Delivery Request slot on an upstream AggDistributionCanal, or the Total Delivery Request slot on a downstream AggDistributionCanal.

3.1.1.2 Sequential Structure

The Sequential Structure models a group of Water Users based on an order of allocation (perhaps by senior water right, etc.). The Agg Diversion Site diverts a bulk amount of water (from a reach or reservoir) and passes it to the first Water User, who will take its requested amount, if possible, and “pass” the remaining water on to the next user. In this way the “junior” users may get only part or none of their requested amount. Excess is returned to the Agg Diversion Site, via the Total Unused Water slot. Each water user processes its delivered amount, and all return flows are handled on a per Water User basis (i.e. each Water User links its returns back to the river individually). It is optional for the user to create a link between Return Flow and the reach or reservoir. If no link is specified, the Return Flow is included in the Incoming Available Water for the next Water User. The Outgoing Available Water from the last Water User (including Return Flow if it is not linked) is linked to the Total Unused Water slot.

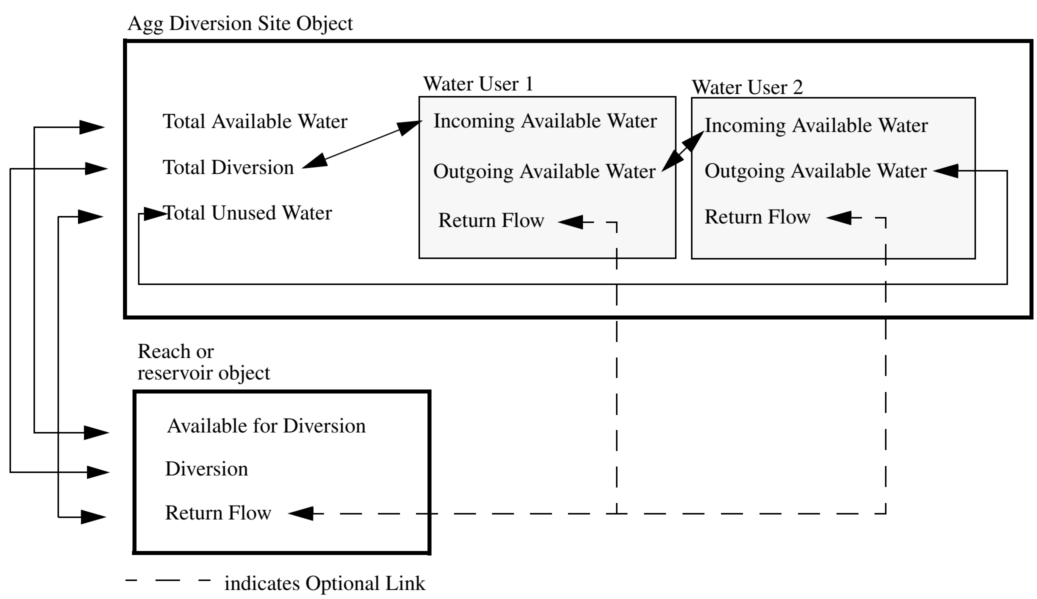


Figure: 2 Sequential Links

Slots Specific to this Method

☞ Total Available Water

Type: Series Slot
Units: FLOW
Description: signifies the water available for diversion from the reservoir or reach to which the diversion is connected.
Information: Set by link propagation from the diverted object. The user is responsible for linking this slot with the object from which the water is diverted.
I/O: Optional; Can be input when it is not linked.
Links: May be linked to the Available For Diversion slot on a Reach object or reservoir, the Outflow slot on a Diversion object, or the Available Flow slot on a Distribution Canal.

☞ Total Depletion

Type: Series Slot
Units: FLOW
Description: sum of the Depletion slots for all Water Users
Information:
I/O: Output only
Links: Usually not linked

☞ Total Depletion Requested

Type: Multislot
Units: FLOW
Description: a multislot showing the Depletion Requested for each Water User as well as the sum of all requested depletions.
Information: Used for all linking structures.
I/O: Optional; Can be input when there are no Water User elements and it is not linked.
Links: Usually not linked

☞ Total Depletion Shortage

Type: Series Slot
Units: FLOW
Description: sum of the Depletion Shortage Slots for all Water Users
Information:
I/O: Output only
Links: Usually not linked

☞ Total Diversion

Type: Series Slot
Units: FLOW
Description: denotes the total water diverted to the Water Users.

Information: Must be linked, by the user, to the Diversion slot on the reach/reservoir being diverted from.
I/O: Optional; Must be input if Total Diversion Requested is not given.
Links: Must be linked to the Diversion slot on a Reach object or reservoir (if it is diverting from one of these objects).

☞ Total Diversion Requested

Type: MultiSlot
Units: FLOW
Description: a multislot showing the Diversion Requested for each Water User as well as the sum of all requested diversions.
Information: Used for all linking structures.
I/O: Optional; Can be input when there are no Water User elements and it is not linked.
Links: May be linked to the Diversion Request slot on a Diversion Object, the Downstream Delivery Request slot on an upstream AggDistributionCanal, or the Total Delivery Request slot on a downstream AggDistributionCanal.

☞ Total Diversion Shortage

Type: Series Slot
Units: FLOW
Description: sum of the Diversion Shortage slots for all Water Users
Information:
I/O: Output only
Links: Usually not linked

☞ Total Unused Water

Type: Series Slot
Units: FLOW
Description: represents the unused water (surface return flow plus any non-diverted water) after all Water Users have processed their requests.
Information: Set by engineering methods after the Water Users dispatch. It is linked, by **RiverWare**™ to the Outgoing Available Water from the last Water User.
I/O: Output only
Links: Return Flow on the object being diverted from.

☞ Total Surface Return Flow

Type: Series Slot
Units: Flow
Description: represents the surface water return flow. It is set equal to the Surface Return Flow of the last element on the aggregate. This is not necessarily the same as the Total Unused Water.

Information: When a Return Flow method is selected, and a Return Flow Split method is selected, and the Link to MODFLOW method are selected the value on the Total Surface Return Flow slot may be transferred through the computational subbasin structure to an external MODFLOW model.
I/O: Output only
Links: Linkable

3.1.1.3 Lumped Structure

The Lumped Structure models a group of Water User's requests as a single lumped sum. The individual requests are summed in the Total Diversion Requested slot on the Agg Diversion Site, and a comparison is made with the Total Available for Diversion slot. If the total request can be met, the Diversion slot is set to that amount; if not, the Diversion slot is set to the Total Available for Diversion amount. In this scheme, the Return Flow is also lumped. The Total Depletion Requested slot tracks the sum of the individual Water User's consumption requests. If a total deficit occurs in the Diversion, the Return Flows are recalculated based on the ratio of the Total Depletion Requested to the Total Diversion Requested. The Total Depletion Shortage is computed as the difference between Total Depletion Request and Depletion. The Total Depletion Shortage is then used to allocate the shortage to the member elements proportional to their depletion request.

It is not necessary to use Water User elements with this linking structure. If desired, Total Diversion Requested and Total Depletion Requested can be input directly by the user. However, it is often desirable to use one or more Water User objects for this purpose.

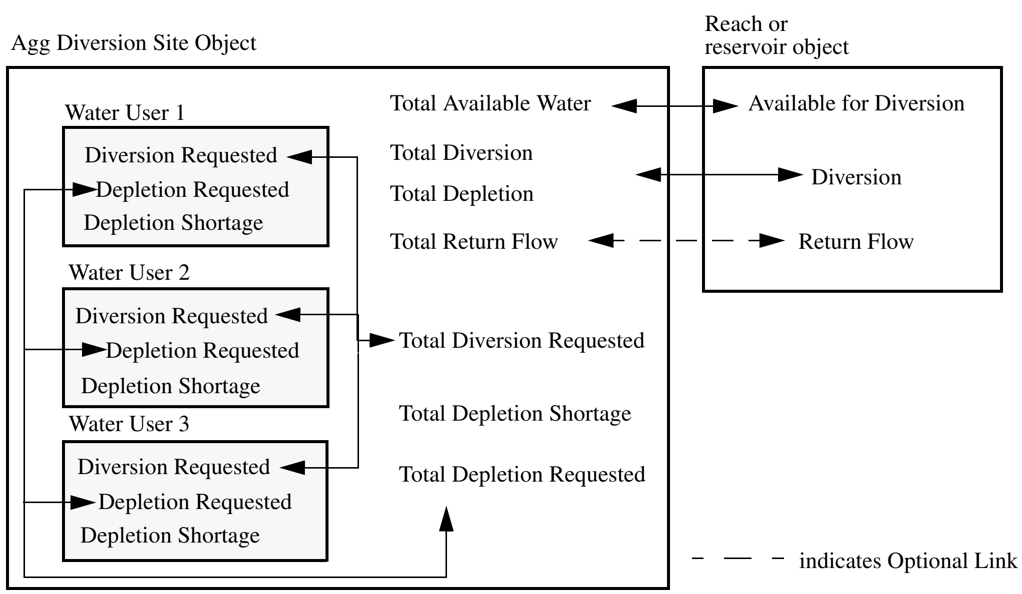


Figure: 3 Lumped Links

Slots Specific to this Method

☞ Total Available Water

Type: Series Slot
Units: FLOW
Description: signifies the water available for diversion from the reservoir or reach to which the diversion is connected
Information: Set by link propagation from the diverted object. The user is responsible for linking this slot with the object from which the water is diverted.
I/O: Optional; Could be input if not set from link.
Links: May be linked to the Available For Diversion slot on a Reach object or reservoir, the Outflow slot on a Diversion object, or the Available Flow slot on a Distribution Canal.

☞ Total Depletion

Type: Series Slot
Units: FLOW
Description: The total amount consumed by the agg.
Information: Computed typically as Total Diversion minus Total Return Flow
I/O: Output only
Links: Usually not linked

☞ Total Depletion Requested

Type: Multislot
Units: FLOW
Description: a multislot showing the Depletion Requested for each Water User as well as the sum of all requested depletions.
Information: Used for all linking structures.
I/O: Optional; Can be input when there are no Water User elements and it is not linked.
Links: Usually not linked

☞ Total Diversion

Type: Series Slot
Units: FLOW
Description: denotes the total water diverted to the Water Users
Information:
I/O: Optional; must be input if Total Diversion Requested is not given.
Links: Must be linked to the Diversion slot of a Reach object or a reservoir (if diverting from one of these objects).

☞ Total Diversion Requested

Type: MultiSlot
Units: FLOW

Description: a multislot showing the Diversion Requested for each Water User as well as the sum of all requested diversions.
Information: Used for all linking structures.
I/O: Optional; Can be input when there are no Water User elements and it is not linked.
Links: May be linked to the Diversion Request slot on a Diversion Object, the Downstream Delivery Request slot on an upstream AggDistributionCanal, or the Total Delivery Request slot on a downstream AggDistributionCanal.

☞ Total Return Flow

Type: Series Slot
Units: FLOW
Description: represents the portion of the water diverted to water users which is not consumed
Information: It is the user's responsibility to specify what is done with this data. If it is not linked to another object (i.e. back to the reach/reservoir Return Flow slot), it is lost from the system.
I/O: Output only
Links: May be linked to the Return Flow slot on any object or the Inflow slot on a Groundwater Storage object.

☞ Total Depletion Shortage

Type: Series Slot
Units: Flow
Description: Represents the difference between Total Depletion Requested and Total Depletion.
Information: Computed as Total Depletion Requested - Total Depletion
I/O: Output Only
Links: NA

3.1.2 Aggregate Conjunctive Use

The Aggregate Conjunctive Use category exists as a user selectable method category only when Lumped Structure is selected as the linking structure. It is used to model supplemental diversions from groundwater. Conjunctive Use calculations also exist for the Sequential Structure and the No Structure, however, the method is selected on the Water User elements for those cases.

3.1.2.1 None

This is the default method. It performs no calculations and there are no slots specifically associated with it.

3.1.2.2 Supplement Diversion Lumped

This method is used to supplement surface water diversions with groundwater. The associated slots are given below.

Slots Specific to this Method

☞ **Available Supplemental Water**

Type: Series Slot
Units: FLOW
Description: represents the amount of water available to supplement surface water diversion
Information:
I/O: Optional; Input if not linked to the Available For Pumping slot in the Groundwater Storage object.
Links: May be linked to Available For Pumping on a Groundwater Storage object.

☞ **Supplemental Diversion**

Type: Series Slot
Units: FLOW
Description: the amount of water diverted from groundwater for supplemental use
Information:
I/O: Output only
Links: May be linked to the Pumped Flow slot on the Groundwater Storage object.

☞ **Total Supplement Return Flow**

Type: Series Slot
Units: FLOW
Description: return flow from the supplemental water
Information:
I/O: Output only

Links: May be linked to the Return Flow slot on any object or the Inflow slot on a Groundwater Storage object.

☞ **TOTAL SUPPLEMENTAL USE REQUEST**

Type: Series Slot
Units: FLOW
Description: the amount of water requested from groundwater for supplemental use
Information: If not input, it is calculated as the difference between Diversion Requested and Diversion.
I/O: Optional
Links: Usually not linked

If Available Supplemental Water is known, the following conjunctive use calculations are performed:

If Total Supplemental Use Request is not input, it is calculated as follows:

$$\text{Total Supplemental Use Request} = MIN(\text{Total Diversion Requested} - \text{Total Diversion}, \text{Maximum Total Supplement Request})$$

(The Maximum Total Supplement Request slot is instantiated when an Aggregate Max Supplemental Request method is selected.)

Supplemental Diversion is then calculated as:

$$\text{Supplemental Diversion} = MIN(\text{Total Supplemental Use Request}, \text{Available Supplemental Water})$$

Total Supplement Return Flow is:

$$\text{Total Supplement Return Flow} = \text{Supplemental Diversion} \times \left(1 - \frac{\text{Total Depletion Requested}}{\text{Total Diversion Requested}}\right)$$

3.1.2.3 Supplement Diversion Sequential

This is not a visible, user-selectable method on the Agg Diversion Site object. It is activated when Supplement Diversion is selected in the Conjunctive Use method category on one or more Water User elements. The slots and calculations associated with this method are given below.

SLOTS SPECIFIC TO THIS METHOD

☞ **AVAILABLE SUPPLEMENTAL WATER**

Type: Series Slot
Units: FLOW
Description: represents the amount of water available to supplement surface water diversion

Information:
I/O: Optional; Input if not linked to the Available For Pumping slot in the Groundwater Storage object.
Links: May be linked to Available For Pumping on a Groundwater Storage object.

☞ SUPPLEMENTAL DIVERSION

Type: Series Slot
Units: FLOW
Description: the amount of water diverted from groundwater for supplemental use
Information: Should be linked to the Pumped Flow slot on the Groundwater Storage object.
I/O: Output only
Links: May be linked to the Pumped Flow slot on the Groundwater Storage object.

☞ TOTAL CONSUMPTION

Type: MultiSlot
Units: FLOW
Description: sum of the Consumption slots of all Water User elements performing conjunctive use calculations
Information: This slot does not consider consumption by Water User elements that do not have Supplement Diversion selected in the Conjunctive Use category.
I/O: Output only
Links: Usually not linked

☞ TOTAL CONSUMPTION SHORTAGE

Type: MultiSlot
Units: FLOW
Description: sum of the Consumption Shortage slots of all Water User elements performing conjunctive use calculations
Information: This slot does not consider consumption shortages of Water User elements that do not have Supplement Diversion selected in the Conjunctive Use category.
I/O: Output only
Links: Usually not linked

☞ TOTAL SUPPLEMENTAL USE REQUEST

Type: Series Slot
Units: FLOW
Description: sum of the Supplemental Use Request slots on all Water User elements performing conjunctive use calculations
Information:
I/O: Output only
Links: Usually not linked

Supplemental Diversion is set as the lower value of either Total Supplemental Use Request or Available Supplemental Water. This represents the amount of water that is available to supplement all the Water User elements' requests. However, there may not be enough water to meet the Supplemental Use Requests of all the elements. In this case, the Agg Diversion Site allocates the supplemental water to the elements based on their Supplemental Diversion Priority (a slot on each element which has Supplement Diversion selected). The element with the highest priority (lowest integer value) is allotted the total amount requested (if possible). If there is supplemental water remaining, it is allotted to the element of second highest priority. This is continued until no supplemental water remains.

3.1.3 Aggregate Max Supplemental Request

This method category is dependent on the Supplement Diversion Lumped method in the Conjunctive Use category. It provides the option of using groundwater storage elevation to determine the maximum supplement request. Values for the Maximum Total Supplement Request slot are then used in Conjunctive Use calculations.

3.1.3.1 None

This is the default method. It performs no calculations and there are no slots specifically associated with it.

3.1.3.2 Input Max Request

When selected, this method instantiates the Maximum Total Supplement Request series slot for user input.

Slots Specific to this Method

☞ **Maximum Total Supplement Request**

Type: Series Slot
Units: FLOW
Description: user input maximum supplemental request
Information:
I/O: Required input
Links: Not linkable

3.1.3.3 GW Elevation Max Request

This method determines the Maximum Total Supplement Request based upon the groundwater elevation at the previous timestep. Values are propagated across a link from the Previous Groundwater Elevation slot on a Groundwater object to the Previous Groundwater Elevation slot on the Agg Diversion Site. An interpolation of these values, using the Max Request Table, provides the Maximum Total Supplement Request.

SLOTS SPECIFIC TO THIS METHOD

☞ PREVIOUS GROUNDWATER ELEVATION

Type: Series Slot
Units: LENGTH
Description: previous groundwater elevation propagated across a link from a Groundwater object
Information:
I/O: Optional; usually set by a link.
Links: Linked to the Previous Groundwater Elevation slot on a Groundwater Storage object.

☞ MAX REQUEST TABLE

Type: TableSlot
Units: LENGTH VS. FLOW
Description: user input related groundwater elevation and maximum supplemental request
Information:
I/O: Required input
Links: Not linkable

☞ MAXIMUM TOTAL SUPPLEMENT REQUEST

Type: Series Slot
Units: FLOW
Description: maximum supplemental request based on the Max Request Table
Information: Maximum Total Supplement Request is calculated through table interpolation of the Max Request Table using the Previous Groundwater Elevation value.
I/O: Output only
Links: Not linkable

3.1.4 MODFLOW Link Category AggDiv

The “MODFLOW Link Category AggDiv” category is used to specify whether the Agg Diversion Site is linked with an external MODFLOW model. TODO LINK

3.1.4.1 No Link to MODFLOW AggDiv

No computations or slots are associated with the “No Link to MODFLOW AggDiv” method. This is the default method for the “MODFLOW Link Category AggDiv”.

3.1.4.2 Link to MODFLOW WU

Note: RiverWare’s connection with MODFLOW is currently not functional. This method has been disabled and cannot be selected. An error will be posted at model load if this method was previously selected. Contact CADSWES for help.

The “Link to MODFLOW WU” method allows the AggDiversion Site object to be linked with MODFLOW. No computations or slots are associated with the “Link to MODFLOW WU” method. A surface water body represented in MODFLOW may receive a surface return flow from a RiverWare AggDiversion Site object. That is, the value of the Total Surface Return Flow slot may be transferred through the computational subbasin structure to MODFLOW. This will only have a non-zero value when a Return Flow method, a Return Flow Split method, and the Link to MODFLOW method are selected on member elements.

Click HERE (Section 7.2.1) to view the “RiverWare - MODFLOW Connection” Functionality Guide . A description of the AggDiversion Site specific data configuration is presented in that guide HERE (Section 7.2.1.3.5).

3.2 Dispatch Methods

No Structure

When the No Structure method is selected, the Agg Diversion Site object only acts as a container for Water User objects. All calculations are performed within the Water Users and links are made directly to the diverted objects. Therefore, the Agg Diversion Site object never dispatches when the user selects No Structure. It does, however, show the Total Diversion Requested and Total Depletion Requested by the Water Users contained within.

Sequential Structure

3.2.1 Process Sequential given Diversion Requested

Required Known Slots

☞ **Total Diversion Requested**

☞ **Total Available Water**

Required Unknown Slots

☞ **Total Diversion**

☞ **Total Depletion**

A general description of the interaction among the Agg Diversion Site, Water Users, and Reach/Reservoir is given below.

1. Reach/Reservoir “Available For Diversion” is set.
2. Value propagates to Agg Diversion Site: “Total Available Water.”
3. **Process Sequential given Diversion Requested** on Agg Diversion Site object dispatches. “Total Diversion” is calculated as the lower value of “Total Available Water” and the amount of water needed to satisfy the Diversion Requested on all Water User elements.

Note: Because the return flows may be available to downstream elements, this value may be less than the Total Diversion Requested.

4. “Total Diversion” propagates to “Incoming Available Water” on the first Water User.

5. The Water User solves for “Diversion”, “Outgoing Available Water” and “Return Flow”.

6. “Return Flow” is propagated to reach/reservoir (if link has been established) or it is sent to the next Water User via the “Outgoing Available Water” slot.

7. “Outgoing Available Water” value propagates to “Incoming Available Water” of second Water User.

8. “Outgoing Available Water” from last Water User propagates to Agg Diversion Site: “Total Unused Water” (including “Return Flow” from the last Water User if it is not linked to the Reach/Reservoir).

9. “Total Surface Return Flow” is set equal to the “Surface Return Flow” on the last element.

10. **Process Sequential given Diversion Requested** on Agg Diversion Site object re-dispatches.

11. “Total Depletion”, “Total Diversion Shortage”, and “Total Depletion Shortage” are calculated.

12. “Total Unused Water” is propagated to the “Return Flow” on the Reach/Reservoir.

13. Conjunctive Use calculations are performed if selected.

3.2.2 Process Sequential given Diversion

This method is executed when Total Diversion is specified by the user. The value of Total Diversion cannot be greater than the Total Available Water.

Required Known Slots

☞ **Total Diversion**

☞ **Total Unused Water**

Required Unknown Slots

☞ **Total Depletion**

A general description of the interaction among the Agg Diversion Site, Water Users, and Reach/Reservoir is given below.

1. The input value of “Total Diversion” propagates to “Incoming Available Water” on the first Water User.

2. The Water User solves for “Diversion”, “Outgoing Available Water” and “Return Flow”.

3. “Return Flow” is propagated to reach/reservoir (if link has been established) or it is sent to the next Water User via the “Outgoing Available Water” slot.

4. “Outgoing Available Water” value propagates to “Incoming Available Water” of second Water User.

5. “Outgoing Available Water” from last Water User propagates to Agg Diversion Site: “Total Unused Water” (including “Return Flow” from the last Water User if it is not linked to the Reach/Reservoir).

6. “Total Surface Return Flow” is set equal to the “Surface Return Flow” on the last element.

7. **Process Sequential given Diversion** on the Agg Diversion Site object dispatches.

8. “Total Depletion”, “Total Diversion Shortage”, and “Total Depletion Shortage” are calculated.

9. “Total Unused Water” is propagated to the “Return Flow” on the Reach/Reservoir.

10. Conjunctive Use calculations are performed if selected.

3.2.3 Process Sequential given Depletion Requested

This method is only used when all of the member elements are modeling Soil Moisture to compute the requests, HERE (Section 27.1.1.5). The member water users compute the Depletion Requested at the beginning of the timestep for the current timestep and some number of future timesteps as described HERE (Section 27.1.1.5). These Depletion Requested values propagate to the aggregate and together with Total Available Water, cause this dispatch method to be used.

Required Known Slots

☞ **Total Depletion Requested**

☞ **Total Available Water**

Required Unknown Slots

☞ Total Diversion

☞ Total Depletion

A general description of the interaction among the Agg Diversion Site, Water Users, and Reach/Reservoir is given below.

1. At the beginning of the timestep, Depletion Requested is set on the member element's slots and is then propagated to the Agg Diversion Site.Total Depletion Requested.

2. Reach/Reservoir "Available For Diversion" is set and the value propagates to Agg Diversion Site.Total Available Water.

3. **Process Sequential given Depletion Requested** on the Agg Diversion Site object dispatches. Total Diversion Requested is computed by solving each water user element without setting any slots. If the previous soil moisture is unknown because it is solving at future timesteps, the soil moisture is assumed to be full so that only Depletion Requests for Crops will be met. (This will iterate once the soil moisture is known) On the aggregate, the "Total Diversion" is calculated as the lower value of "Total Available Water" and the amount of water needed to satisfy the Diversion Requested on all water user elements.

Note: Because the return flows may be available to downstream elements, this value may be less than the Total Diversion Requested.

4. "Total Diversion" propagates to "Incoming Available Water" on the first Water User.

5. The Water User solves for "Diversion", "Outgoing Available Water" and "Return Flow".

6. "Return Flow" is propagated to reach/reservoir (if link has been established) or it is sent to the next Water User via the "Outgoing Available Water" slot.

7. "Outgoing Available Water" value propagates to "Incoming Available Water" of second Water User.

8. "Outgoing Available Water" from last Water User propagates to Agg Diversion Site: "Total Unused Water" (including "Return Flow" from the last Water User if it is not linked to the Reach/Reservoir).

9. "Total Surface Return Flow" is set equal to the "Surface Return Flow" on the last element.

10. **Process Sequential given Depletion Requested** on the Agg Diversion Site object re-dispatches.

11. "Total Depletion", "Total Diversion Shortage", and "Total Depletion Shortage" are calculated.

12. "Total Unused Water" is propagated to the "Return Flow" on the Reach/Reservoir.

13. Conjunctive Use calculations are performed if selected.

LUMPED STRUCTURE

3.2.4 Process Lumped given Diversion Requested

REQUIRED KNOWN SLOTS

☞ **TOTAL DIVERSION REQUESTED**

☞ **TOTAL AVAILABLE WATER**

REQUIRED UNKNOWN SLOTS

☞ **TOTAL DIVERSION**

☞ **TOTAL DEPLETION**

A general description of the interaction among the Agg Diversion Site, Water Users, and Reach/Reservoir is given below.

1. Reach/Reservoir "Available For Diversion" is set.

2. This value propagates to Agg Diversion Site: "Total Available Water."

3. **Process Lumped given Diversion Requested** dispatches on the Agg Diversion Site object.

4. Within the dispatch method, "Total Diversion" and "Total Return Flow" are calculated.

5. The values propagate to the reach or reservoir.

6. Conjunctive Use and Aggregate Max Supplemental Request calculations are performed if selected.

If Total Depletion Requested is not specified, it is set equal to the Total Diversion Requested.

If Total Diversion Requested is greater than the Total Available Water:

$$\text{Total Diversion} = \text{Total Available Water}$$

$$\text{Total Return Flow} = \text{Total Diversion} \times \left(1 - \frac{\text{Total Depletion Requested}}{\text{Total Diversion Requested}}\right)$$

If the Total Diversion Requested is less than Total Available Water:

$$\text{Total Diversion} = \text{Total Diversion Requested}$$

$$\text{Total Return Flow} = \text{Total Diversion} - \text{Total Depletion Requested}$$

Then, Total Depletion is set as the difference between Total Diversion and Total Return Flow.

Total Depletion Shortage is computed as Total Depletion Requested minus Total Depletion.

The totalDepletionFraction is computed as Total Depletion divided by Total Depletion Requested. This value is not shown to the user. The agg then loops over each member water user (WU) and computes and sets the element's Depletion Shortage. It is computed as

$$\text{WU.Depletion Shortage} = \text{WU.Depletion Requested} \times (1 - \mathit{totalDepletionFraction})$$

Essentially, any shortage is shared by each water user element proportional to its Depletion Requested. Click **HERE (Depletion Shortage)** for more information.

When the **Specify Scheduled Requests** (**HERE (Specify Scheduled Requests)**) is selected on the element, additional shortage terms are computed as follows:

$$\text{WU.Depletion Schedule Cutback[ ]} = \text{WU.Depletion Schedule[ ]} - \text{WU.Depletion Requested[ ]}$$

$$\text{WU.Depletion Schedule Shortage[ ]} = \text{WU.Depletion Schedule Cutback[ ]} + \text{WU.Depletion Shortage[ ]}$$

3.2.5 Process Lumped given Diversion

REQUIRED KNOWN SLOTS

☞ **TOTAL DIVERSION**

REQUIRED UNKNOWN SLOTS

☞ **TOTAL DEPLETION**

A general description of the interaction among the Agg Diversion Site, Water Users, and Reach/Reservoir is given below.

1. **Process Lumped given Diversion** dispatches on the Agg Diversion Site object.

2. Within the dispatch method, “Total Depletion” and “Total Return Flow” are calculated.

3. The values propagate to the reach or reservoir.

4. Conjunctive Use and Aggregate Max Supplemental Request calculations are performed if selected.

If Total Depletion Requested is not specified, it is set equal to the Total Diversion Requested.

If Total Diversion is less than the Total Diversion Requested:

$$\text{Total Return Flow} = \text{Total Diversion} \times \left(1 - \frac{\text{Total Depletion Requested}}{\text{Total Diversion Requested}}\right)$$

If the Total Diversion is equal to the Total Diversion Requested:

$$\text{Total Return Flow} = \text{Total Diversion - Total Depletion Requested}$$

Then, Total Depletion is set as the difference between Total Diversion and Total Return Flow.

Total Depletion Shortage is computed as Total Depletion Requested minus Total Depletion.

The totalDepletionFraction is computed as Total Depletion divided by Total Depletion Requested. This value is not shown to the user. The agg then loops over each member water user element and computes and sets the element's Depletion Shortage. It is computed as:

$$\text{WaterUser.Depletion Shortage} = \text{WaterUser.Depletion Requested} \times (1 - totalDepletionFraction)$$

Essentially, any shortage is shared by each water user element proportional to its Depletion Requested. Click HERE (Depletion Shortage) for more information.

When the **Specify Scheduled Requests** (HERE (Specify Scheduled Requests)) is selected on the element, additional shortage terms are computed as follows:

$$\text{WU.Depletion Schedule Cutback[ ]} = \text{WU.Depletion Schedule[ ]} - \text{WU.Depletion Requested[ ]}$$

$$\text{WU.Depletion Schedule Shortage[ ]} = \text{WU.Depletion Schedule Cutback[ ]} + \text{WU.Depletion Shortage[]}$$

4. Aggregate Reach

The Aggregate Reach (Agg Reach) is an aggregate of Reach Objects. The aggregate performs no routing calculations, it just totals various information from its elements. It also handles the connection between elements. This is done by linking inflow of the aggregate object to the inflow of the top element, then the outflow of the top element to the inflow of the next, and so on, until the last object's outflow is connected to the Agg Reach's outflow.

General slots

(slots which always appear for this object)

☞ INFLOW

Type: SeriesSlot

Units: FLOW

Description: inflow into the top of the aggregate

Information: It is automatically linked to the inflow of the first element in the list of elements, if any.

I/O: Optional; Depends on desired dispatching.

Links: May be linked to the Outflow of any object.

☞ OUTFLOW

Type: SeriesSlot

Units: FLOW

Description: outflow from bottom of aggregate

Information: It is automatically linked to the outflow of the last element in the list of elements, if any.

I/O: Optional; Depends on desired dispatching.

Links: May be linked to the Inflow of any object.

☞ TOTAL DIVERSION

Type: SeriesSlot

Units: FLOW

Description: total amount of water diverted from elements within the Aggregate Reach
Information: This is calculated in the dispatch method.
I/O: Output only
Links: Usually not linked

☞ Total Local Inflow

Type: SeriesSlot
Units: FLOW
Description: total amount of local inflow into the elements within the Aggregate Reach
Information: This is calculated in the dispatch method.
I/O: Output only
Links: Usually not linked

☞ Total Return Flow

Type: SeriesSlot
Units: FLOW
Description: total amount of water returned to elements within the Aggregate Reach
Information: This is calculated in the dispatch method.
I/O: Output only
Links: Usually not linked

4.1 User Methods

This object has no user methods.

4.2 Dispatch Methods

This object has only one dispatch method.

4.2.1 solveTotals

This dispatch method totals various auxiliary slots of the elements.

REQUIRED KNOWNS:

- ☞ **INFLOW**
- ☞ **OUTFLOW**

REQUIRED UNKNOWNS:

- ☞ **TOTAL DIVERSION**
- ☞ **TOTAL RETURN FLOW**
- ☞ **TOTAL LOCAL INFLOW**

When the inflow and the outflow of the Aggregate Reach are known, it is assumed that all interior reaches have dispatched, and have solved for all auxiliary slots. This method then sums all these values (Diversion, Return Flow, and Local Inflow) and reports them in the appropriate total slot on the Aggregate Reach. If any value is a NaN, the total will not include any contribution from that slot of that element

5. Bifurcation

This object models a bifurcation (division into two or more branches) of water.

General Slots

☞ Inflow

Type: Multislot

Units: Flow

Description: the general flows into the bifurcation object, which can be split

Information: This multislot is a general slot used by the user methods: Two Outflows and Fractionally Split Outflows.

I/O: Optional: can input, set by propagation across a link, set via rules, or calculated (only if the Two Outflows method is selected). Inflow must be input, set via link propagation, or by rules if the Fractionally Split Outflows method is selected.

Links: The multislot (first column of the slot) may be linked to the Outflow of any object.

5.1 User Methods

5.1.1 Bifurcation Outflow

This method category allows the user to specify the number of destinations to which outflow can be sent.

5.1.1.1 Two Outflows

The Two Outflows method is the default method. The Outflow1 and Outflow2 slots are specific to the Two Outflows method. No calculations are actually performed by this user method. It dictates that the object can dispatch with the following methods only: solveInflow, solveOutflow1, or solveOutflow2.

☞ **OUTFLOW1**

Type: AggSeriesSlot
Units: FLOW
Description: outflow from one side of the bifurcation
Information:
I/O: Optional: can be input, set via rules, set by propagation across a link, or calculated
Links: May be linked to the Inflow of any object.

☞ **OUTFLOW2**

Type: AggSeriesSlot
Units: FLOW
Description: outflow from the other side of the bifurcation
Information:
I/O: Optional: can be input, set via rules, set by propagation across a link, or calculated
Links: May be linked to the Inflow of any object.

5.1.1.2 Fractionally Split Outflows

Fractionally Split Outflows splits the Outflow to two or more destinations. Two slots are specific to the Fractionally Split Outflows method: Outflow Fractions Table and Split Outflows. The Split Outflows slot is a noComputeMultislot allowing the user to split outflows to multiple destinations. Each of the multiple outflows is represented by a subslot in the Split Outflows multislot. A subslot is created automatically each time the Split Outflows slot is linked to a destination object. The Outflow Fractions Table contains one column per subslot on the Split Outflows multislot. Each column contains the fraction multiplier dictating what fraction of the total outflow goes to each destination.

The number of outflows solved for depends on the number of links the user has specified from the Split Outflows noComputeMultislot to destination objects.

The multiple outflows are calculated as:

$$\text{Split Outflows (i)} = \text{Outflow Fractions (i)} \times \text{Inflow} \quad \text{Eq. 1}$$

where *i* is the column of the Outflow Fractions Table. The first column on the Split Outflows multislot is the multislot itself and is not one of the subslots. It is set as the sum of all the subslot values. Therefore it should be equal to the total outflow from the Bifurcation object. The following slots are instantiated when this method is selected:

☞ OUTFLOW FRACTIONS TABLE

Type: TableSlot

Units: FRACTION

Description: a table giving the proportion of outflow going to each Split Outflows subslot.

Information: The values in this slot should add up to 1.0. For example, if there are two columns and the inflow is split evenly, 0.5 would go into one column and 0.5 would go into the second column. The table will have the same number of columns as the number of links to the Inflow multislot. The columns are automatically added and named whenever a link is added to the Split Outflows noComputeMultislot.

I/O: Required Input

Links: Not linkable.

☞ SPLIT OUTFLOWS

Type: NoComputeMultislot

Units: FLOW

Description: the multiple (or split) outflows based on the Outflow Fractions Table.

Information: The subslots of this multislot contain the multiple outflows. A subslot is created when the user links this slot to a destination object.

I/O: Output only

Links: linkable to multiple slots

5.1.2 Bifurcation Solution Direction

This method category is used to specify the direction in which the Bifurcation solves (downstream given inflow and one outflow or the outflow fractions; upstream given the outflows). For basic Simulation runs direction does not make a difference and the default method should remain selected. However, in Rulebased Simulation the user may need to limit the Bifurcation to a downstream direction only. If necessary, the Bifurcation can be forced to solve for only Outflow1 or Outflow2 using the appropriate method.

5.1.2.1 Solve Upstream or Downstream

The Solve Upstream or Downstream method is the default user method. If this method is active, it allows all dispatch methods to be active. No slots or calculations are specifically associated with this method.

5.1.2.2 Solve Downstream Only

This method allows the Bifurcation object to only solve in the downstream direction with the following dispatch methods: solveOutflow1, solveOutflow2, or solveSplitOutflows. No slots are specifically associated with this method.

5.1.2.3 Solve Outflow1 Only

This method allows the Bifurcation object to only solve for Outflow1 using the solveOutflow1 dispatch method. No slots are specifically associated with this method.

5.1.2.4 Solve Outflow2 Only

This method allows the Bifurcation object to only solve for Outflow2 using the solveOutflow2 dispatch method. No slots are specifically associated with this method.

5.2 Dispatch Methods

This object solves for inflow, outflow1, outflow2, or splitOutflows depending on the knowns and unknowns.

5.2.1 solveInflow

Solves for Inflow. This dispatch method is available only if the Solve Upstream or Downstream and Two Outflows user methods are active.

REQUIRED KNOWNS

☞ **OUTFLOW1**

☞ **OUTFLOW2**

REQUIRED UNKNOWNS

☞ **INFLOW**

The equation for this dispatch method is:

$$Inflow = Outflow1 + Outflow2$$

5.2.2 solveOutflow1

Solves for Outflow1. This dispatch method is available only if the Two Outflows user method and one of the following is active: Solve Upstream or Downstream, Solve Downstream Only, or Solve Outflow1 Only.

REQUIRED KNOWNS

☞ **INFLOW**

☞ **OUTFLOW2**

REQUIRED UNKNOWNS

☞ **OUTFLOW1**

The equation for this dispatch method is:

$$Outflow1 = Inflow - Outlow2$$

5.2.3 solveOutflow2

Solves for Outflow2. This dispatch method is available only if the Two Outflows user method and one of the following is active: Solve Upstream or Downstream, Solve Downstream Only, or Solve Outflow2 Only.

REQUIRED KNOWNS

☞ **INFLOW**

☞ **OUTFLOW1**

REQUIRED UNKNOWNS

☞ **OUTFLOW2**

The equation for this dispatch method is:

$$Outlow2 = Inflow - Outflow1$$

5.2.4 solveSplitOutflows

Sets the Split Outflows calculated by the Fractionally Split Outflows method. This dispatch method is specifically associated with the Fractionally Split Outflows user method. It does not depend on the bifurcation solution direction.

REQUIRED KNOWNS

☞ **INFLOW**

☞ **OUTFLOW FRACTIONS TABLE**

REQUIRED UNKNOWNS

☞ **SPLIT OUTFLOWS**

This dispatch method sets the multislot and subslots of the Split Outflows noComputeMultislot. Actual calculations are performed by the user method Fractionally Split Outflows.

6. Canal

The canal object models gravity flow through a canal connected to two Reservoirs. The set of allowable dispatch methods for the linked reservoirs is limited. Click **HERE (Allowable Dispatch Methods for Linked Reservoirs)** for details about which dispatch methods are allowed.

General slots

(slots which always appear for this object)

☞ ELEVATION 1

Type: AggSeriesSlot
Units: LENGTH
Description: Pool Elevation of reservoir 1
Information: If the Barkley Canal Flow method is selected, Reservoir 1 must be Barkley, and this slot must be connected to the Pool Elevation on that Reservoir.
I/O: Set through a link
Links: Must be linked to the Pool Elevation slot on a reservoir.

☞ ELEVATION 2

Type: AggSeriesSlot
Units: LENGTH
Description: Pool Elevation of reservoir 2
Information: Set through a link
I/O: Must be linked to the Pool Elevation slot on a reservoir.

☞ FLOW 1

Type: AggSeriesSlot
Units: FLOW
Description: flow to or from the canal end connected to Reservoir 1
Information: If the Barkley Canal Flow method is selected, Reservoir 1 must be Barkley, and this slot must be connected to the Canal Flow slot on that Reservoir.
I/O: Output only
Links: Must be linked to the Canal Flow slot on a reservoir.

☞ FLOW 2

Type: AggSeriesSlot
Units: FLOW
Description: flow to or from the canal end connected to Reservoir 2
Information: Must be connected to the Canal Flow slot on a reservoir.

I/O: Output only
Links: Must be linked to the Canal Flow slot on a reservoir.

6.1 User Methods

6.1.1 Canal Flow

This category controls how the flow through the canal is calculated. Two of the user methods in this category are implementations of canal flow equations used by TVA. The Canal Flow Table method is the only generic method that can be used for canals other than the Tellico and Barkley Canals in the TVA system.

6.1.1.1 None

This method is the default for this category. It will result in an error if it is selected and a run is begun.

There are no slots specifically associated with this method.

6.1.1.2 Canal Flow Table

This method is a general method that can be used to solve any canal whose flow can be described using the Canal Flow Table. This method takes the average pool elevation of the lower Reservoir and the difference in average elevation between the two Reservoirs and uses linear interpolation to determine Canal Flow from the Canal Flow Table.

SLOTS SPECIFIC TO THIS METHOD

☞ **DELTA ELEVATION**

Type: AggSeriesSlot
Units: LENGTH
Description: The average elevation difference between the two reservoirs for the current timestep
I/O: Output only
Links: Not Linkable

☞ **LOWER ELEVATION**

Type: SeriesSlot
Units: LENGTH
Description: The average elevation of the lower reservoir for the current timestep
Information: This slot is necessary because the height of the water in the canal effects the flow rate.
I/O: Output Only
Links: Not Linkable

☞ **CANAL FLOW TABLE**

Type: TableSlot

Units: LENGTH, LENGTH, FLOW
Description: 3-D table used to find canal flow by interpolation
Information: 1st column is Lower Elevation (can have several entries with the same lower elevation), 2nd column is Delta Elevation, 3rd column is Canal Flow. This table provides information regarding several possible Lower Elevation and Delta Elevation values and the corresponding Canal Flow at these operating points.
I/O: Input Only
Links: Not Linkable

☞ **ELEVATION 1 ADJUSTMENT**
Type: ScalarSlot
Units: LENGTH
Description: Adjusts the Elevation 1 value by a user specified amount.
Information: Automatically set to zero if not input. This slot was added to handle the Barkley Canal calculations. This slot will not be necessary for most users.
I/O: Optional; defaults to zero if not input.
Links: Not Linkable

Note: The canal flow solution requires that the slot minimum and maximum be set in the Pool Elevation slot configuration on both linked reservoirs. Also the quality of the solution for Canal Flow is sensitive to the slot convergence values in the slot configuration for all series slots on the Canal object. It is recommended that the slot convergence be set to 0.00001 Percent for all Canal series slots.

First, the method checks to see if an Elevation 1 Adjustment value is given. If no value is input, Elevation 1 Adjustment is set to zero. Otherwise, the Elevation 1 value is adjusted by the given amount.

Then Delta Elevation is computed as:

abs(average elevation 1 - average elevation 2)

Lower Elevation is then set to the lower value of either average elevation 1 or average elevation 2.

Once Delta Elevation and Lower Elevation are known, the Canal Flow can be found directly from the Canal Flow Table. The appropriate direction of flow will be determined based on the average elevation of the two reservoirs. A negative flow implies flow out of the Reservoir. Flow will obviously be from the reservoir with the higher average elevation to the reservoir with the lower average elevation.

6.1.1.3 Tellico Canal Equation

There are no slots specifically associated with this method.

Note: The canal flow solution requires that the slot minimum and maximum be set in the Pool Elevation slot configuration on both linked reservoirs. Also the quality of the solution for Canal Flow is sensitive to the slot convergence values in the slot configuration for all series slots on the Canal object. It is recommended that the slot convergence be set to 0.00001 Percent for all Canal series slots.

The basic equation for this method is in user units:

$$FlowLower = (44000 + 4730(hlower - 805))\sqrt{hdiff}$$

- ***FlowLower* -** Flow into the reservoir with the lower pool elevation (in cfs).
- ***hlower* -** Pool Elevation of the lower reservoir (in feet).
- ***hdiff* -** Absolute value of difference in Pool Elevation between the two reservoirs (in feet).

If the difference in Pool Elevation is smaller than 0.01 feet, a linearized version of this equation is used:

$$FlowLower = (44000 + 4730(hlower - 805))(\sqrt{0.01})\frac{hdiff}{0.01}$$

6.1.1.4 Barkley Canal Equation

There are no slots specifically associated with this method.

Note: The canal flow solution requires that the slot minimum and maximum be set in the Pool Elevation slot configuration on both linked reservoirs. Also the quality of the solution for Canal Flow is sensitive to the slot convergence values in the slot configuration for all series slots on the Canal object. It is recommended that the slot convergence be set to 0.00001 Percent for all Canal series slots.

The basic equation for this method is in user units:

$$FlowLower = (hlower - 335)(1285.1 + 44.07(hlower - 335))\sqrt{hdiff}$$

- ***FlowLower* -** Flow into the reservoir with the lower pool elevation (in cfs).
- ***hlower* -** Pool Elevation of the lower reservoir (in feet).
- ***hdiff* -** Absolute value of difference in Pool Elevation between the two reservoirs (in feet).

If the difference in Pool Elevation is smaller than 0.01 feet, a linearized version of this equation is used:

$$FlowLower = (hlower - 335)(1285.1 + 44.07(hlower - 335))(\sqrt{0.01})\frac{hdiff}{0.01}$$

Barkley must be connected to the Reservoir 1 slots, Flow 1 and Elevation 1. This requirement is due to an adjustment of -0.08 feet made to Elevation 1 because of physical considerations.

6.2 Dispatch Methods

6.2.1 solveFlow

This dispatch method finds the flow through the canal.

In order for the canal to dispatch correctly, the linked reservoirs must solve with one of a limited set of dispatch methods. The table below lists which dispatch methods are allowed for reservoirs linked to a canal and which are not. If a reservoir linked to a canal tries to dispatch with a method that is not allowed, the run will abort with an error message.

Allowable Dispatch Methods for Linked Reservoirs

Dispatch Methods Allowed for Reservoirs Linked to a Canal	Dispatch Methods *NOT* Allowed for Reservoirs Linked to a Canal
solveMB_givenInflowHW	solveMB_givenEnergyInflow
solveMB_givenOutflowHW	solveMB_givenInflowRelease
solveMB_givenInflowStorage	solveMB_givenEnergyStorage
solveMB_givenOutflowStorage	solveMB_givenEnergyHW (Slope Power Reservoir)
solveMB_givenInflowOutflow	solveMB_givenInflowOutflowHW (Storage Reservoir)
solveMB_givenInflowOutflowHW (Level Power Reservoir only)	solveMB_givenInflowOutflowStorage (Storage Reservoir)
solveMB_givenInflowOutflowStorage (Level Power Reservoir only)	
solveMB_givenEnergyHW (Level Power Reservoir only)	

Required Knowns

- ☞ **Elevation 1**
- ☞ **Elevation 2**

Required Unknowns

☞ Flow 1

☞ Flow 2

The dispatch method of the canal object is unique, because the canal solves the reservoirs and the canal at the same time. The usual dispatching pattern is unstable when three objects are iterating with each other. If the Pool Elevations or Storages on both Reservoirs are input, the canal flow can be immediately found. If the reservoirs are solving for Storage/Pool Elevation, the canal performs a modified bisection algorithm to solve for Canal Flow. Following is a description of how these objects interact and how the method solves.

Note: The canal flow solution requires that the slot minimum and maximum be set in the Pool Elevation slot configuration on both linked reservoirs. Also the quality of the solution for Canal Flow is sensitive to the slot convergence values in the slot configuration for all series slots on the Canal object. It is recommended that the slot convergence be set to 0.00001 Percent for all Canal series slots.

When a reservoir dispatches to solve for Pool Elevation, if it is linked to a canal, it sets its Pool Elevation slot with a seed value (previous value) and exits its dispatch method. It will wait for the canal to solve and set the Canal Flow; then it will dispatch again. When both reservoirs have set their Pool Elevation slots, the canal's solveFlow dispatch method goes on the queue and executes. The result is the flow in the canal. When the flow is set on Flow1 and its negative value is set on Flow2, the values propagate to each reservoir. Each reservoir re-dispatches and solves mass balance to set the Storage and Pool Elevation. Note that the seed values in the Pool Elevation slots are only used to cause the canal to dispatch. They are not actually used in the calculation.

The canal flow algorithm begins by summing side inflows to both reservoirs. These include Hydrologic Inflow, Diversion, Return Flow, Inflow Forecast and Pumped Storage Flow. Any of these that are not in use default to zero. The side inflows do not change based on the Canal Flow and thus are treated as a constant in the canal flow algorithm.

Note: Bank Storage, Evaporation, and Seepage are not included in the Canal Flow solution as they can be a function of Pool Elevation. Any non-default method selection in these three categories will result in an error at the start of the run.

Then Canal Flow is set to an initial estimate of zero. With an estimate for Canal Flow, an estimate can be calculated for the remaining 8 variables (Outflow, Inflow, Storage and Pool Elevation on both reservoirs) in the following order:

Note: The following description uses the syntax ObjVariable to indicate that these values are the intermediate values used in the calculation, not the values on the slots. If a variable is known, then ObjVariable is set equal to Object.Slot. Otherwise, it must be computed within the iterative bisection algorithm. Also, a positive canal flow represents flow from Res1 to Res2 and a negative canal flow represents flow from Res2 to Res1.

Res1Outflow: If not known, then Res1.Inflow must be known, and the outflow is either a function of maximum outflow or mass balance. If Res1.Outflow is flagged Max Capacity (M), then Res1Outflow is:

$$Res1Outflow = getMaxOutGivenIn(\text{Res1Inflow,Res1SideFlows})$$

The function getMaxOutGivenIn uses the known Inflow, either input or propagated across a link from an upstream object, to compute the maximum outflow using the specified spill and power methods.

If Res1.Outflow is not flagged max capacity, then Res1.Storage must be input (or Res1.Pool Elevation is input and thus storage is known). Then Res1.Outflow is calculated from the mass balance:

$$Res1Outflow = \text{Res1.Inflow} + canalFlowEst + Res1SideFlows - \frac{\text{Res1.Storage} - \text{Res1.Storage(-1)}}{Timestep}$$

Res1Inflow: If not known, then Res1.Outflow must be known and Res1.Storage must be input (or Res1.Pool Elevation is input and thus storage is known). Then Res1.Inflow is calculated from the mass balance:

$$Res1Inflow = \frac{\text{Res1.Storage} - \text{Res1.Storage(-1)}}{Timestep} + \text{Res1.Outflow} - canalFlowEst - Res1SideFlows$$

Res1Storage: If not known, the Storage is a calculated from the mass balance:

$$Res1Storage = \text{Res1.Storage(-1)} + Res1Inflow - Res1Outflow + canalFlowEst + Res1SideFlows$$

Res1Pool Elevation: If not input, the pool elevation depends on the type of reservoir. For sloped reservoirs, the selected slope storage calculation is executed. It is a function of inflow, outflow and storage:

$$Res1PoolElevation = f(\text{Res1Inflow , Res1Outflow, Res1Storage})$$

If it is not a sloped reservoir, the Pool Elevation is looked up on the Elevation Volume table:

$$Res1PoolElevation = f(\text{Res1Storage})$$

Then the same variables are calculated for Reservoir 2 with the one difference being that a positive canal flow represents a flow out of the reservoir.

Res2Outflow: If not known, then Res2.Inflow must be known, and the outflow is either a function of maximum outflow or mass balance. If Res2.Outflow is flagged Max Capacity (M), then Res2Outflow is:

$$Res2Outflow = getMaxOutGivenIn(\text{Res2Inflow,Res2SideFlows})$$

The function getMaxOutGivenIn uses the known Inflow, either input or propagated across a link from an upstream object, to compute the maximum outflow using the specified spill and power methods.

If Res2.Outflow is not flagged max capacity, then Res2.Storage must be input (or Res2.Pool Elevation is input and thus storage is known). Then Res2.Outflow is calculated from the mass balance:

$$Res2Outflow = \text{Res2.Inflow} - canalFlowEst + Res2SideFlows - \frac{\text{Res2.Storage} - \text{Res2.Storage(-1)}}{Timestep}$$

Res2Inflow: If not known, then Res2.Outflow must be known and Res2.Storage must be input (or Res2.Pool Elevation is input and thus storage is known). Then Res2.Inflow is calculated from the mass balance:

$$Res2Inflow = \frac{\text{Res2.Storage} - \text{Res2.Storage(-1)}}{Timestep} + \text{Res2.Outflow} + canalFlowEst - Res2SideFlows$$

Res2Storage: If not known, the Storage is a calculated from the mass balance:

$$Res2Storage = \text{Res2.Storage(-1)} + Res2Inflow - Res2Outflow - canalFlowEst + Res2SideFlows$$

Res2Pool Elevation: If not input, the pool elevation depends on the type of reservoir. For sloped reservoirs, the selected slope storage calculation is executed. It is a function of inflow, outflow and storage:

$$Res2PoolElevation = f(\text{Res2Inflow , Res2Outflow, Res2Storage})$$

If it is not a sloped reservoir, the Pool Elevation is looked up on the Elevation Volume table:

$$Res2PoolElevation = f(\text{Res2Storage})$$

With two new pool elevations, a new Canal Flow can be calculated based on the user-selected Canal Flow method.

$$canalFlowUpdated = f(Res1PoolElevation, Res2PoolElevation)$$

For the first iteration, with an estimate of zero canal flow, the resulting pool elevations represent the maximum possible elevation difference and thus produce a *canalFlowUpdated* that is the maximum possible canal flow (absolute value). This represents an upper bound on the magnitude of canal flow, with the initial estimate of zero as the lower bound. These two bounds are then bisected to give a new estimate:

$$canalFlowEst = \frac{canalFlowEst + canalFlowUpdated}{2}$$

Then the process of calculating updated values for all variables is repeated. Each iteration provides two new bounds for canal flow, which move toward convergence with each iteration. In some cases, *canalFlowUpdated* can fall outside of the current bounds. In these cases, the algorithm can determine whether *canalFlowEst* for that iteration was too low or too high. It then replaces *canalFlowUpdated* with the appropriate existing bound when bisecting to calculate a new estimate. In other words, the solution will never allow the bounds to expand from one iteration to the next. They will always be converging.

CONVERGENCE AND MAX ITERATIONS WITHIN CANAL SOLUTION:

Following are the convergence criteria used in the canal flow algorithm. That is, after each iteration, the new value is compared to the previous computed value. The iteration is stopped when the difference between old and new are within the indicated criteria. The iterations will also stop the run with an error if max iterations has been exceeded.

Item	Stopping Criteria
Canal Flow	10 m^3/s
Reservoir Storage	1000 m^3
Reservoir Inflows and Outflows	10 m^3/s
Reservoir Pool Elevations	0.01 m
Max Iterations	100

The quality of the canal flow solution is dependent on the quality of data in the Canal Flow Table and the reservoir tables. Inaccurate data in these tables or insufficient precision in these data may cause the solution to have difficulty converging.

7. Computational Subbasin

The Computational Subbasin supports computations that involve more than one object, so-called "global" solutions, though they need not encompass the entire network. A Computational Subbasin is created through the same dialogs as other typed subbasins, but it, like simulation objects, contains slots and other attributes that you can inspect and change, by "opening" the subbasin. For more information on creating, deleting, and viewing computational subbasins, click **HERE (Subbasins.pdf, Section 1)** to go to the subbasin documentation. Computational subbasins contain no general slots; all slots are dependent on user-selectable methods.

Selected slot values may be propagated from a computational subbasin to *all* member simulation objects. When you choose to propagate a slot named "S", the value of the slot is copied to the slot named "S" on all objects in the subbasin. Propagation has no effect on member objects that do not contain a slot named "S".

The subbasin may also be "verified". The meaning of this depends on the selected methods. During verification, RiverWare performs checks that can be made prior to a run and that might discover errors before a run is begun, so that such errors can be identified and corrected early.

The subbasin is either "enabled" or "disabled". When enabled, it will be verified at the beginning of a run, and if its verification fails, the run will abort. If disabled, it will not be verified at the beginning of a run. This allows subbasin verification not to interfere with model runs while the model or subbasin is under development.

General Slots

☞ **NONE**

7.1 User Methods

7.1.1 Diversions from Reservoirs

7.1.1.1 None

There are no slots or calculations associated with this method. This method cannot be selected (i.e. the user must select a method other than “None”) if the ComputeReservoirDiversions, HERE (RPLPredefinedFunctions.pdfRPLPredefinedFunctions.pdf, Section 20), RPL function is being used.

7.1.1.2 Operating Level-Based

This method is used in conjunction with the ComputeReservoirDiversions RPL function. For more information on the ComputeReservoirDiversions function, click HERE (RPLPredefinedFunctions.pdfRPLPredefinedFunctions.pdf, Section 20),

This method is used to meet multiple Water User demands via reservoir diversions. Each reservoir can supply one or more Water Users and each Water User may divert from one or more reservoirs. The ComputeReservoirDiversions function computes, for each Water User object, the portion of water supplied by each connected reservoir. This information is set on the Supply From Reservoirs slot on the Water User object. The data can then propagate from the Water User to the reservoir Diversion slot via Diversion Objects (see diagram below). The details of the calculations are included in the help file for the ComputeReservoirDiversions RPL function.

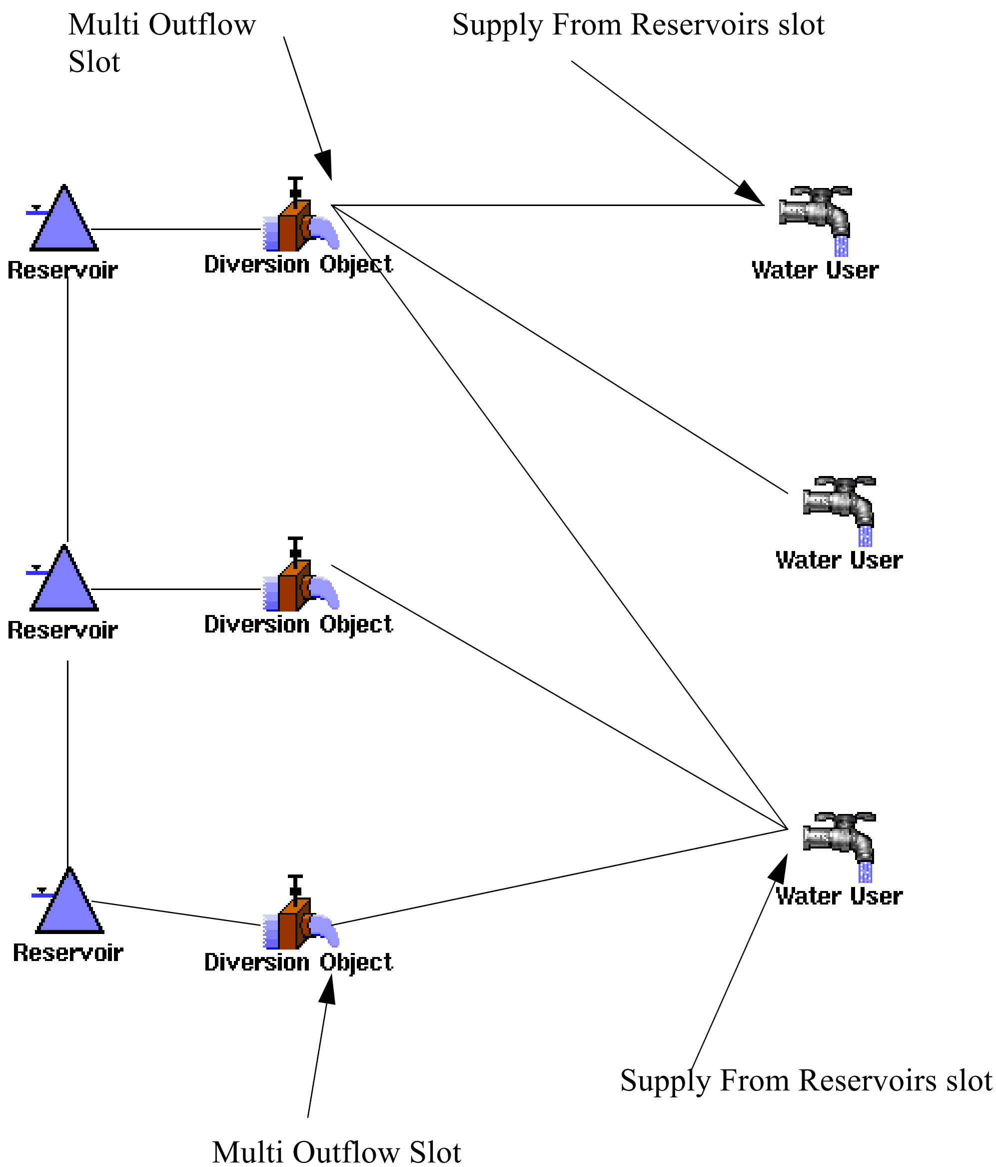
Use of this method for USACE-SWD is described HERE (USACE_SWD.pdf, Section 3.8).

SLOTS SPECIFIC TO THIS METHOD

☞ **BOTTOM OF CONSERVATION POOL**

Type: Scalar Slot
Units: NONE
Description: The Operating Level that represents the bottom of the conservation pool
Information: The operations/calculations associated with this method only apply to the conservation pool. If a reservoir is below the bottom of the conservation pool, it is not considered for diversions.
I/O: Required input
Links: Not linkable

The use of this method and the ComputeReservoirDiversions RPL function requires a specific configuration of objects and method selections. The schematic diagram below displays the required object and link configuration.



In the above diagram, the Diversion slot on each reservoir is linked to the Diversion slot on the Diversion Object. The demands are represented by the Water User objects. The Supply From Reservoirs slot on each Water User is linked to the Multi Outflow slot on each Diversion Object that can act as a supply for that demand. The rule sets the values on the Supply From Reservoirs slots. These propagate to the Multi Outflow slots on connected Diversion Objects. The Diversion objects solve for their Diversion slot. The Diversion values are passed to the Diversion slot on the Reservoir object and the water is removed from the Reservoir. On each reservoir, the Conservation and Flood Pools method in the

Operating Levels category should be selected to instantiate the Bottom of Conservation Pool slot.

7.1.2 Low Flow Releases

7.1.2.1 None

There are no slots or calculations associated with this method. This method cannot be selected (i.e. the user must select a method other than “None”) if the “MeetLowFlowRequirement”, HERE (RPLPredefinedFunctions.pdfRPLPredefinedFunctions.pdf, Section 120), RPL function is being used.

7.1.2.2 Operating Level-Based

This method is used in conjunction with the “MeetLowFlowRequirement” RPL function. For more information on the “MeetLowFlowRequirement”, click HERE (RPLPredefinedFunctions.pdfRPLPredefinedFunctions.pdf, Section 120).

This method is used to meet Control Point low flow requirements with releases from reservoir. Each reservoir can supply some water to the Control Point. The “MeetLowFlowRequirement” function computes a release from each reservoir in a Control Point’s list. The details of the calculations are included in the help file for the “MeetLowFlowRequirement” RPL function. Use of this method for USACE-SWD is described HERE (USACE_SWD.pdf, Section 3.7).

Slots Specific To This Method

☞ **Bottom of Conservation Pool**

Type: Scalar Slot
Units: NONE
Description: The Operating Level that represents the bottom of the conservation pool
Information: The operations/calculations associated with this method only apply to the conservation pool. If a reservoir is below the bottom of the conservation pool, it is not considered for low flow releases.
I/O: Required input
Links: Not linkable

7.1.3 Flood Control

The Flood Control methods calculate a flood control release for the current simulation day. The two flood control methods, Operating Level Balancing and Phase Balancing, are invoked from a predefined rules function, and their results are returned to the calling rule.

For USACE-SWD methods, the approach is described as described **HERE (USACE_SWD.pdf, Section 3.6)**. The flood control methods (other than None) use forecast data (storage, inflows, empty space at 0 through Forecast Period timesteps after the current simulation timestep). They propose release schedules for each reservoir for each timestep in the forecast period. They do this to account for routing effects. They return the values for the first timestep of the proposed schedule (for the current simulation timestep).

A sample rule for using flood control with a computational subbasin named “Flood Basin” is as follows:

FOREACH (LIST triplet IN "FloodControl“(“Flood Basin”)) DO

(triplet <0>) [] = triplet <1>

ENDFOREACH;

This rule invokes the predefined RPL function “FloodControl”, **HERE (RPLPredefinedFunctions.pdfRPLPredefinedFunctions.pdf, Section 40)**, which returns a list of {slot, value, object} triplet. The rule iterates over the list, assigning the value (index 1 from triplet) to the slot (index 0 from triplet). The rules function returns three {slot, value, object} triplets for each reservoir:

{*res*.Flood Control Release, *value*, object}, {*res*.Outflow, *value*, object}, and for Operating Level Balancing method {res.Target Balance Level, value, object}

Thus, three slots may be set for each reservoir in the subbasin. The value for the Outflow slot is the sum of the Surcharge Release, Flood Control Release and the Flood Control Minimum Release. The Outflow slot is a dispatch slot, so setting this slot will cause the reservoir to dispatch and the outflows to be propagated downstream. If no flood operations are required, the values returned for Flood Control Release and Outflow will be zero, and each reservoir in the subbasin will dispatch no outflow. The Target Balance Level is the value as assigned by one or more key control point.

If a flood control method determines that flood operations are needed when the end of the run is within Forecast Period timesteps, the method will issue a warning suggesting that the user extend the post-run dispatching past the time when flood operations are needed. The method will then return zeroes for the values in the {slot, value} pairs. For more information on post-run dispatching, click **HERE (RunControl.pdf, Section 2.2)**. Typically, you can set the **Number of Post-Run Dispatch Timesteps** equal to the Forecast Period.

In the event that flood operations are needed and forecast data are missing, but the end of the dispatching is not within Forecast Period timesteps, the method will issue an error message and the run will abort.

The flood control methods are computationally intensive and make assumptions about the model configuration. Each method makes different assumptions, which are described below, in the context of the method. Checks are performed at the beginning of the run to catch errors that would cause the flood control to fail. These checks (also performed through the

GUI “verify” button on the computational subbasin’s Open Object dialog) allow a user to correct errors early.

At the time these checks are made, the subbasin creates topological indices, that is, maps that cache the downstream and upstream relationships that are used frequently during the computationally-intensive flood control algorithms. These indices persist throughout the run, so any changes to the topology during a run will cause undefined, possibly disastrous results. These indices contain only the objects of interest to the flood control methods, which are reservoirs and control points. Reaches, stream gages, water users, and other such objects are ignored. The relationships among the reservoirs and control points are determined by following the “main channel outflow” slot links, described in the following table:

Table 1. Main Channel Slots

Object Type	Main Channel Outflow Slot
AggDistributionCanal	Total Outflow
Agg Diversion Site	Total Return Flow
Bifurcation	Outflow1
DiversionObject	(none)
StreamGage	Gage Outflow
WaterUser	(none)
all others	Outflow

By performing the checks and building the indices only at the beginning of the run, the computational cost of running the flood control method is reduced.

Note: Because these checks are not in the critical path of the flood control algorithm, *the user must not modify the model during a run.* The only GUI operations that are permissible during a flood-control run are read-only operations, such as plotting and viewing data.

RiverWare’s behavior is undefined (may crash) if a model is modified during a flood control run.

Flood control uses linear routing coefficients as a computationally-cheap way to approximate the routing that occurs during simulation. Proper choice of routing coefficients is essential to producing high-quality results with the flood control methods.

Note: Flood control optimizes the control points’ channel space using the linear routing coefficients on the control point; however, the Reaches may use a non-linear routing method in the simulation. The closer the linear routing coefficients approximate the routing methods on the Reaches, the better the flood control algorithm will work. If the linear routing coefficients are bad approximation of the routing methods used on the Reaches, the flows routed in the simulation may be vastly different than the flows approximated by the flood control algorithm used to be release decisions. As a result, the simulated flows may not optimize channel capacity and oscillations in the flood control releases may result.

Note: Flood control assumes that flood control releases will be routed to future time steps. Therefore, the Impulse Response routing method should not be used with this flood control method.

General Beginning-of-Run Checks

The subbasin must be contiguous and must contain no loops.

7.1.3.1 None

None is the default for the category, and will result in an error if it is selected and a run calls the rules “FloodControl” method on the subbasin. (Disabling the subbasin does not disable the rule -- it simply disables the verification process at the beginning of the run.)

Slots Specific to this Method

☞ **None**

7.1.3.2 Phase Balancing

Beginning-of-run Checks Specific to this Method

The slots Forecast Phases and Forecast Period must be valid (have defined values).

All reservoirs in the Upstream Reservoirs slot of a key control point must have the Phase Balancing method selected in the Flood Control Release category.

All propagable slots, when also instantiated on members, must match those of the subbasin.

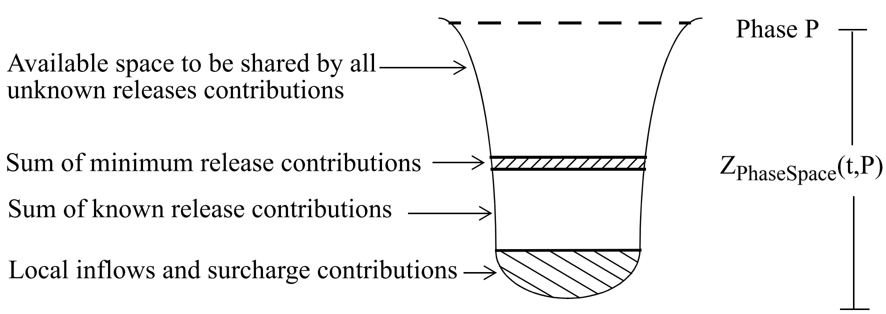
Behavior of this Method

The method calculates flood control release values at the current timestep for each reservoir in the subbasin by the following steps:

1. Set the proposed flood control release slot of each reservoir in the subbasin for each timestep in the forecast period to the reservoir’s maximum allowable release. The reservoir’s maximum allowable releases for each timestep in the forecast period is calculated using the reservoir’s objective release pattern, which is then constrained by the reservoir’s permissible outflow change tables and the maximum limit of the outlet works. The objective release pattern attempts to evacuate the flood control storage (including forecasted inflows) in the number of timestep specified by the pattern from the first unconstrained release. If the volume of water is constrained by the reservoir's constraints or by downstream control points within the objective pattern threshold the release pattern should be maintained. If the objective patter threshold is surpassed the objective release pattern is reapplied starting with the current timestep. This value is based on the reservoir's objective release pattern, the reservoir's max release slot and the reservoir's permissible outflow constraints slots. This release value is the target release, which will likely be constrained by control points downstream.

2. For each phase (phase III down to phase I) and for each timestep calculate the phase space allocations at each control point:

- Calculate the reservoir's phase space allocation at the current control point. The phase space allocation at each control point is calculated by determining which reservoir releases contribute to the water in the control point. The contributing reservoir releases are determined from the control point's reservoir list and the linear routing coefficients. Available space in the control point is calculated by taking the phase space hydrograph and subtracting out local inflows, all contributions of known reservoir releases, and minimum release contributions from reservoirs whose release is unknown. Known reservoir releases are considered releases that occurred before t - lag time from the reservoir to the control point. These are releases that have already been constrained by the current control point. Unknown reservoir releases are all reservoir releases occurring at or after t - lag time from the reservoir to the control point. These are releases that have not yet been constrained by the current control point. The available space is then divided up to all unknown reservoir releases which will contribute the control point at the current timestep. The space is divided among reservoirs using the reservoirs' linear routing coefficients and reservoirs' lake character weight at the timestep t-1-lag time from the reservoir to the control point.



Control Point Z channel at t

The lake character value is calculated by the reservoir as follows: The Lake Character is a weighting factor that will be determined for certain types of Reservoir Objects (Storage and Level Power Reservoirs) to be used as a means to balance all reservoirs in the same Operating Level. Once the Lake Character is known, it can be used in conjunction with other reservoirs of the same Operating Level to divvy up the available empty space at downstream control points. A reservoir's Lake Character for each timestep is calculated by:

$$LakeCharacter = Coefficient \times PercentageFloodPoolCurrentlyOccupied$$

The coefficient is either the value in the Lake Character Coefficient scalar slot or the value in the Variable Lake Character Coefficient series slot if input for that timestep. Riverware will check at each timestep if a value has been input in the Variable Lake Character slot. If valid, the Coefficient is the input value for some timestep from Variable Lake Character Coefficient. When NaN, the Coefficient is the value in the Lake Character Coefficient ScalarSlot.

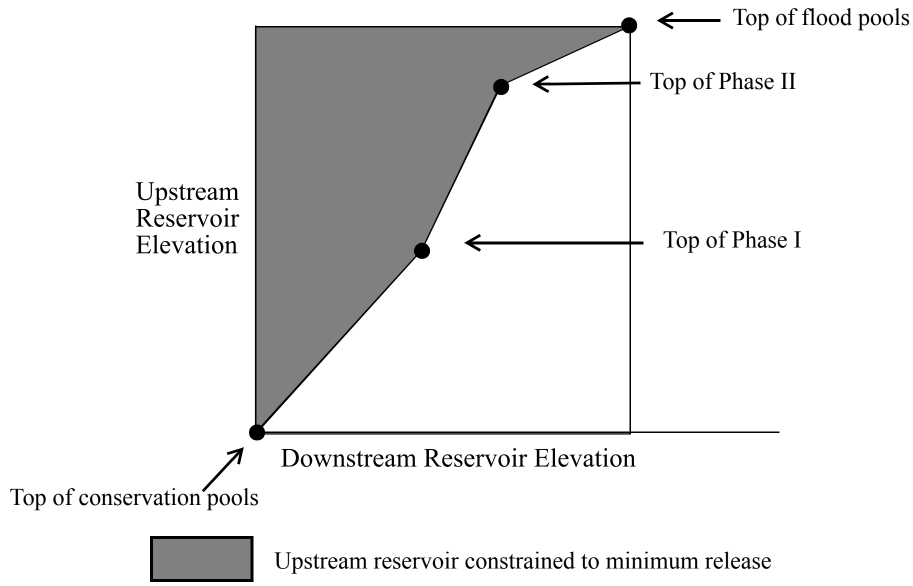
The percentage of the flood pool that is currently occupied is calculated as

$$\frac{Storage - Storage(TopofConservationPoolLevel)}{Storage(TopofFloodPoolLevel) - Storage(TopofConservationPoolLevel)}$$

This storage value takes into account the Surcharge Release determined for the current timestep. The Top of Flood Control Pool Level and Top of Conservation Pool Level are required input. The storage corresponding to these level will be determined from an interpolation of the Elevation-Volume TableSlot.

- The proposed flood control release slot for each reservoir is then set to the minimum of the current value of the proposed flood control release or the phase space allocation at the control point divided by the linear coefficient. Dividing the allocation by the linear coefficient results in a release value that will wholly fill the allocated space. Recall the minimum release contribution was subtracted out of the available phase space, ensuring that space would be available for all minimum releases.
- If the control point lists a reservoir directly upstream (that is, there is no time lag from the reservoir to the control point) and the reservoir has upstream tandem reservoirs, the control point may constrain upstream tandem reservoirs to their minimum release. Two reservoirs which are located so that a release from one becomes the inflow of the other are in tandem. The upstream reservoir releases will be set to the reservoir's minimum release if the two reservoirs are out of *tandem balance*. A tandem balancing curve is constructed for each pair of tandem reservoirs by interpolating straight lines between the following points calculated from the tandem operating levels table slot and the tandem operating aberrations slot: a) top of flood pool; b) the top of Phase II; c) the top of Phase I; d) the top of the conservation pool. If more water is stored in the downstream tandem reservoir than indicated by the tandem balance curve then the temporary release from the

upstream reservoir will be set the minimum release. Given a reservoir D and its tandem



upstream reservoir U. The tandem upstream reservoir is constrained to its minimum release if the point ($D_{PoolElevation}$(t-1), $U_{PoolElevation}$(t-1-lagTime)) lies within the shaded area. If the point lies in the unshaded area, the flood control release is the same as the parallel case.

- If not all of the control point's available phase space was taken by its reservoirs' releases (one or more of the reservoirs may have been constrained upstream) the phase space allocation is repeated with all reservoirs which are not constrained upstream, considering all releases constrained upstream as *known* releases. This process is repeated until the entire phase space is allocated for the current timestep or until all of the current control point's reservoirs are found to be constrained upstream.

Note: Please read the notes regarding the flood control methods in general, above.

SLOTS SPECIFIC TO THIS METHOD

☞ FORECAST PERIOD

Type: Scalar Slot
Units: NONE
Description: Timesteps in the period for which forecast data are available.
Information: Minimum value of 1.
I/O: Required Input.
Links: May not be linked.

☞ **TOP OF CONSERVATION POOL**

Type: Scalar Slot
Units: NONE
Description: Operating level associated with the top of the conservation pool of every reservoir in the subbasin.
Information: Also known as "target operating pool level, since this is the preferred, or target, level for all reservoirs. This level is also the bottom of the flood pool.
I/O: Required Input.
Links: May not be linked.

☞ **TOP OF FLOOD POOL**

Type: Scalar Slot
Units: NONE
Description: Operating level associated with the top of the flood pool.
Information: Must be above the top of the conservation pool.
I/O: Required Input.
Links: May not be linked.

☞ **NUMBER OF PHASES**

Type: Scalar Slot
Units: NONE
Description: The number of phases associated with the all the reservoirs in the subbasin.
I/O: Required Input.
Links: May not be linked.

7.1.3.3 Operating Level Balancing

This method uses a series of passes over successively lower operating levels (called "balance levels" in this context), in which it attempts to reduce all the reservoirs in the subbasin to the operating level of the pass. It attempts to release as much water as feasible as soon as possible. A set of criteria is applied, reducing the potential release from each full reservoir so that

- flooding does not occur downstream at the control points to which this reservoir routes[1],
- water is not released from the reservoir's conservation pool, and the release schedule does not rely on any water being released from the conservation pool in its projections,
- priority is given to reservoirs based on their operating levels ("fullness"),
- the reservoirs subject to a key control point are left as balanced[2] as possible, given the above,

1. except when permitted at a control point by means of a flooding exception (method selection) at that control point.
2. with a minimal spread in operating levels

- flood pools are drained as soon as possible and within the forecast period, given the above, and
- reservoirs have a release schedule that is as smooth as possible, given the above.

A more detailed description of the operation of this method follows the list of slots below.

When this method is selected, dependent method categories Balance Level Determination, Pass Behavior, Key Control Point Max Release, Key Control Point Space Use, Operating Level Mapping, Tandem Balancing, Modify Inputs to Two-Reservoir Midpoint, Tandem Storage Management, Tandem Storage Considered, Priority Determination, Reservoir Set, Smoothing Releases, Last Pass Timesteps May Increase apply, and Control Point Variable Routing Coefficients. They are described below.

Note: The Operating Level Balancing method was developed under the aegis of the U.S. Army Corps of Engineers - Southwest Division, with the expressed purpose of approximating algorithms and results of portions of the Corps' SUPER suite of programs. Some of the methods described below (among those dependent on Operating Level Balancing) exist solely for the testing and acceptance phase of this project. Some of these methods and the slots associated with them may not persist in RiverWare, and are so indicated below. Other methods exist for the purpose of experimentation, to enable interested persons to study the effects and interactions of various algorithmic details. The defaults were chosen to match the SUPER behavior. Use of this method for USACE-SWD is described HERE (USACE_SWD.pdf, Section 3.6).

Note: Undocumented temporary slots may appear (with "Temp" or "Temporary" in their names) in various releases of RiverWare, and may disappear later, for the reasons stated in the above note. Such temporary slots are not stored with a model, and are not deemed available to users for use in rules, user-defined accounting methods, or for any other purpose. In RiverWare 6.1, many of the Temp slots were converted to regular slots but then made invisible. These invisible slots but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).

Slots Specific to this Method

☞ Forecast Period

Type: Scalar Slot
Units: NONE
Description: Timesteps in the period for which forecast data are available.
Information: Minimum value of 1. May be propagated from the subbasin to its members. All subbasin members with this slot must have the same value in the slot.
I/O: Required Input.
Links: May not be linked.

☞ Balance Period

Type: Scalar Slot
Units: NONE
Description: Timestep in the future (origin 1, i.e., 1 means the current simulation timestep) at which the forecast storage in the flood pool determines the amount of water to be released over the forecast period by the flood control algorithm.
Information: Minimum value of 1. Less than or equal to the value of the Forecast Period. Values must match on all objects in the subbasin that have this slot. May be propagated from the subbasin to its members.
I/O: Required Input.
Links: May not be linked.

☞ Top of Conservation Pool

Type: Scalar Slot
Units: NONE
Description: Operating level associated with the top of the conservation pool of every reservoir in the subbasin (all reservoirs in the subbasin must have the same value for this slot).
Information: Also known as “target operating pool level”, since this is the preferred, or target, level for all reservoirs. This level is also the bottom of the flood pool. May be propagated to every subbasin member that has this slot.
I/O: Required Input.
Links: May not be linked.

☞ Top of Flood Pool

Type: Scalar Slot
Units: NONE
Description: Operating level associated with the top of the flood pool of every reservoir in the subbasin (all reservoirs in the subbasin must have the same value for this slot).
Information: Must be greater than the Top of Conservation Pool. Used to compute surcharge releases. May be propagated to every subbasin member that has this slot.

☞ Highest Operating Level

Type: Scalar Slot
Units: NONE
Description: The highest operating level that is valid for every reservoir in the subbasin. The Operating Level Balancing method will not consider operating levels higher than this when it attempts to balance reservoirs.
Information: Used in the Two-Reservoir Midopoint method of the Tandem Balancing category (dependent on the Operating Level Balancing method selection). Every reservoir in the subbasin must have an Operating Level Table whose

domain covers this operating level because the Two-Reservoir Midpoint method looks up storages associated with levels up to and including this.

I/O: Required Input.

Links: May not be linked.

☞ LOWEST OPERATING LEVEL

Type: Scalar Slot

Units: NONE

Description: The lowest operating level that is valid for every reservoir in the subbasin. The Operating Level Balancing method will not consider operating levels lower than this in its attempts to balance reservoirs.

Information: Used in the Two-Reservoir Midopoint method of the Tandem Balancing category (dependent on the Operating Level Balancing method selection). Every reservoir in the subbasin must have an Operating Level Table whose domain covers this operating level. The Two-Reservoir Midpoint method looks up storages associated with this balance level on each reservoir in the subbasin.

I/O: Required Input.

Links: May not be linked.

☞ ROUTED FLOW TOLERANCE

Type: Scalar Slot

Units: FLOW

Description: Performance tuning knob. Flow, due to a routed flood control release, below which it is not worth the computational expense to continue processing.

Information: When, during the computation of a prospective additional flood control release, the resulting routed flow at a control point is below this value, the method considers it to be zero, at (and downstream of) this control point.

I/O: Required input. Default value 0.000001 cms.

Links: May not be linked.

☞ INCREMENTAL RELEASE TOLERANCE

Type: Scalar Slot

Units: FLOW

Description: Performance tuning knob. Magnitude of additional release (for a pass of the flood control algorithm) below which it is not worth the computational cost to consider.

Information: When, during the computation of a prospective additional flood control release, the additional release becomes limited to a magnitude below this, the additional release is taken to be 0.0.

I/O: Required Input. Default value 0.000001 cms.

Links: May not be linked.

BEGINNING-OF-RUN CHECKS SPECIFIC TO THIS METHOD

Prior to the start of a run, the subbasin is analyzed and verified, at which time the following checks are made:

1. On the subbasin:

- The following slot values must be positive: Forecast Period, Balance Period, Highest Operating Level.
- The following slot values must be non-negative: Routed Flow Tolerance, Lowest Operating Level.
- The Balance Period must be less than or equal to the Forecast Period.
- The Highest Operating Level must be greater than the Lowest Operating Level.

2. On member reservoirs:

- The following slot values must be positive: Allowable Rising Release Change, Allowable Falling Release Change, Maximum Release Variation.
- The following slot values must be non-negative: all elevations in the Operating Level Table.
- For each date in the Operating Level Table, the elevations must monotonically increase with the operating level, and the Operating Level Table must cover the range Lowest Operating Level through Highest Operating Level (from the subbasin slots).
- Each reservoir must have the Operating Level Balancing method selected for the Flood Control Release category.
- (This is not a check, but an action taken:) A slot, Operating Level Storage Table, is created at this time. It is a direct mapping between operating levels and volumes, created from the Elevation-Volume Table and the Operating Level Table. For performance purposes, this table is used by the flood control and related methods in lieu of two table interpolations for mapping storage volumes to operating levels and back. Also, see the Conditional Operating Level category (HERE (Objects.pdf, Section 22.1.19) for more information on when this slot could be re-computed.

3. On member control points:

- The following slot values must be positive: Excepted Releases, if the control point has a flooding exception.
- The following slot values must be non-negative: elements of Routing Coefficients.
- If the control point is key, its Key Control Point Balancing method must be Operating Level Balancing.

- The sum of the elements of the Routing Coefficients vector for each Upstream Reservoir must be within Routed Flow Tolerance of 1.0.
- The Empty Space slot must be valid, which means that a method other than None for Regulation Discharge must be selected. This slot determines how much water may flow at a control point without causing flooding. Flooding is defined as flow exceeding the regulation discharge.
- Each control point must have the Operating Level Balancing method selected for the Flood Control Release category.

4. Relationships among slot values of objects in the subbasin:

- The following slots, when instantiated on members, must match those of the subbasin: Balance Period, Forecast Period, Top of Conservation Pool, Top of Flood Pool.

5. Relationships among objects in the subbasin:

- Each reservoir has a member control point “immediately” (considering only reservoirs and control points) downstream, which is called its “output gage control point”. The Routing Coefficients vector from the reservoir to its output gage control point is exactly { 1.0 }.
- All subject reservoir of a key control point (those named in the control point’s Key Control Point Reservoirs slot) must be members of the subbasin and upstream of the control point.
- When the selected method for Key Control Point Space Use is Key Control Point Balancing Share, there may not be non-subject reservoirs interposed between a key control point and any of its subject reservoirs. This is because the non-subject tandem may take on storage of water released from a subject reservoir upstream, but be unable to release it. When the selected method is First-Come First-Served, it doesn’t matter, as all reservoirs have equal access to the key control point’s empty space.
- Every reservoir in a control point’s Upstream Reservoirs slot must be a subbasin member and must be upstream of the control point.
- Note that there is no beginning-of-run check that the Key Control Point Reservoirs are among the Upstream Reservoirs, but the Key Control Point Balancing method Operating Level Balancing will issue a fatal error if they are not.

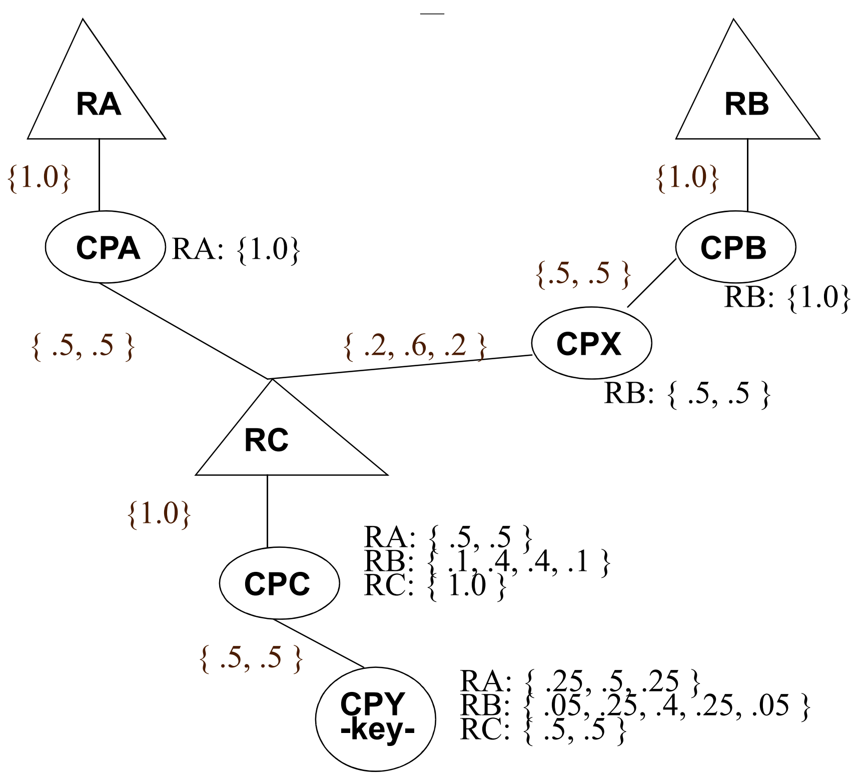
Note: If a given control point has the Compute Aggregate Coefficients method selected in the Variable Routing Coefficients category, the temp Routing Coefficients slot will be used in place of the Routing Coefficients slot in all calculations. This slot is populated at the beginning of the timestep based on the flows in the system. See the Compute Aggregate Coefficients method documentation in the Control Point Variable Routing Coefficients category below.

Example Subbasin

For illustrating the behavior of this method with examples, we refer to the following simplified subbasin (with only reservoirs and control points extracted; Reaches and other objects are not of interest to the flood control algorithm and so are not considered in any discussion of the flood control algorithm). This example subbasin, illustrated below, contains three reservoirs, RA, RB and RC. Each reservoir has an “output gage control point”, CPA, CPB, CPC, respectively. Two more control points, CPX and CPY are in the subbasin. In the subbasin from which this is extracted, there would be Reaches between CPB and CPX, and between CPC and CPY. There would also be a confluence above RC, possibly with Reaches between it and CPA and between it and CPX. During simulation, the dispatching of reaches would affect the routing. which is incremental (one reach at a time), but during execution of the flood control rule, the linear routing coefficients on the control points are used to approximate the composite (cumulative) routing between reservoirs and control points.

For the purpose of illustration, we assume that this subbasin has

- a daily timestep,
- a forecast period of 5days,
- a balance period of 3 days,
- the top of the conservation pool at operating level 5.0,
- hypothetical units, in which 1 unit of volume == 1 unit of flow (for simplicity)
- (hidden) reaches with routing coefficients as depicted below.



Behavior of this Method

The Operating Level Balancing method first determines if flood control is needed. If no reservoirs in the subbasin are forecast to be in the flood pool on the current simulation timestep, it returns with no flood control releases for any of the reservoirs in the subbasin.

Note: If a reservoir in the computational basin is disabled and is set to Pass Inflows as described **HERE (Objects.pdfObjects.pdf, Section 24.1.30.2)**, the reservoir is not considered in the computations below. The reservoir will pass the flows through including any upstream releases that could possibly have been held as tandem storage.

If flood control is needed, the method initializes the flood control release schedule[1] for each reservoir to zero: no flood control release on any timestep of the forecast period. It then invokes the selected Key Control Point Balancing method on each of the key control points in the subbasin. After this, the method selects a set of balance levels to use to drive the algorithm. This set is determined by the selected method in the Balance Level Determination

1. By “schedule”, we mean a proposed release for each timestep in the forecast period. Only the first timestep’s release will be simulated after the flood control rule function returns, but the flood control algorithm proposes a schedule for the entire forecast period to attain smoothness in release hydrographs and to respect priorities in the presence of routing.

category (q.v., below). The set is sorted in descending order (duplicate values are removed) and preserved so it can be returned to the calling rule/function to set on the Target Balance Level slot.

The flood control method then runs one or more "passes" over the reservoirs in the subbasin. There is one pass for each balance level in the above-mentioned set, and an additional "final" pass associated with the operating level that is the Top of Conservation Pool. The goal of each pass is to determine the maximum release that each reservoir can make on each timestep of the forecast period, to bring its forecast operating level to the balance level of the pass as soon as possible[1]. No attempt is made to reduce a reservoir below the level of the pass.

At the beginning of each pass, the method determines the set of "full" reservoirs, namely those whose forecast operating levels (at the end of the balance period) exceed the balance level of the pass [2]. The set of full reservoirs is sorted by fullness (fullest one first).

The full reservoirs are considered in order, and for each full reservoir there are two steps to compute a release schedule. Remember that all the actions described here are on intermediate (temporary or invisible) slots; the simulation state of objects is not changed by the flood control method. (Empty space and storage slots are not changed.)

1. Undo the proposed releases from the prior pass (if any), freeing space at control points and removing tandem storage at tandem reservoirs. This step is not necessary if accumulating additional releases on each pass (Pass Behavior method selected is Compute Additional Release).

2. Loop over the timesteps in the forecast period, computing a tentative release for each timestep. In this loop are two steps:

- Compute a tentative release for the full reservoir for the forecast timestep in question, based on empty space at the downstream control points, on our ability to store water in a downstream tandem reservoir, and on constraints of the full reservoir; and
- "Apply" that tentative release, updating the empty space at downstream control points and storages at downstream reservoirs.

1. The method determines a reservoir's excess water, VOL, which is the volume of water in the reservoir at the end of the balance period that is above the balance level of the pass. The goal is to release VOL over the forecast period, but as soon as possible. There is no guarantee that the entire VOL will be released by the end of the balance period.
2. As the reader will see, user-selectable methods, described below, modify the meaning of a "forecast operating level at the end of the balance period". For the purpose of this discussion, think of it as the forecast storage at the end of the balance period as computed before the flood control method was called, plus any water proposed for release from an upstream reservoir on a prior pass and arriving for storage within the balance period, minus water proposed for release within the balance period on any prior pass. Methods that modify its meaning include Operating Level Mapping, Tandem Balancing, Modify Inputs to Two-Reservoir Midpoint, Tandem Storage Management, Tandem Storage Considered.

Example: RA is forecast to be at level 10.2, RB at 8.9, RC at 9.5. The key control point, CPY, assigned balance level 9.0 to reservoirs RA, RB and RC, and so its balance level for the current timestep is 9.0. The balance level set is { 9.0, 5.0 }. There are three passes, one at 9.0, one at 5.0 and a “final” pass at 5.0, in which certain constraints, described below, are removed. On the first pass, the algorithm proposes release schedules for RA and RC that will bring them down as close to operating level 9 as possible in 5 days, given empty space constraints at control points. It considers RA before RC because RA is forecast to be “fuller” (at a higher operating level). On the second pass, the algorithm proposes releases for RA, RC, and RB, to bring them down to operating level 5.0 while still respecting all the constraints of the first pass. On the final pass, all reservoirs are considered, with the goal of bringing them to level 5.0, with slightly different constraints, as described below.

Downstream Objects that Constrain a Full Reservoir’s Release

The objects that constrain a reservoir’s release are those reachable by water routed within the forecast period. For the purpose of the flood control method, water travels only as far as

- there are control points with routing coefficients from that full reservoir, and
- it can travel within the forecast period, and
- it will result in at least Routed Release Tolerance cms on the timestep in question. (Note that this is internal units.)

Example: If there were no routing coefficients at CPC for routing from RA to CPC, the flood control algorithm would stop checking constraints at CPA, even if control points downstream of CPC had routing coefficients for releases from RA. If the routing coefficients for RA to CPC were { 0,0,0,.5,.5 }, no releases after the 2nd day would be considered from CPC on downstream, because no water from such releases would reach CPC or beyond within the forecast period. Finally, if the Routed Release Tolerance were .000001 cms, and a release were to route to a control point but the arriving flow were to be less than .000001 cms, the flood control algorithm would stop at said control point. Thus, the Routed Release Tolerance can be used to tune the algorithm - to make it run faster, but with less precision.

The flood control algorithm does not check for consistency of the linear routing coefficients with the simulation routing. If the linear routing coefficients used by the flood control do not approximate the routing used in simulation, results may be of low quality. Similarly, if the routing coefficients at two control points are inconsistent, RiverWare will not notice.

Example: Routing coefficients from RA to CPC could be {.5,.5} and from RA to CPY could be {.8,.2}, and RiverWare would not notice. It is entirely up to the modeler to see that the coefficients are consistent with each other and with the simulation routing methods.

Example: The routing coefficients on the Reach between CPA and RC might be {.1,.2,.3,.4}, and all else as described in the reference subbasin. Riverware would use the routing coefficients on the reach during simulation, and the routing coefficients on the control points for flood control. The fact that they don’t match would not be discovered by RiverWare. The Reaches might not use linear routing at all. RiverWare would have no way to tell if the linear routing used in flood control is an “acceptable” approximation of the (possibly non-linear) routing selected on the Reaches and used during simulation.

Empty Space Restrictions at Control Points

Limited channel (empty) space at downstream control points may reduce the proposed release. The available empty space on a pass is determined as follows:

- On all passes, all empty space at non-key control points is available on a first-come, first-served basis.
- Space at key control points is reserved for use by subject full reservoirs on passes prior to the final pass. This allocation of space is respected on these passes, unless overridden by the selected method in the Key Control Point Space Use category (q.v., below). See below for a discussion of the allocation of water stored in downstream tandem reservoirs.
- On the final pass, all empty space at all control points that has not been used heretofore is available on a first-come, first-served basis.

Example: The key control point balancing method on CPY apportioned its empty space as follows: 60% to RA and 40% to RB. With three passes, at operating levels 9.0, 5.0 and 5.0, on the first two passes (9.0 and 5.0), RA releases can consume at most 60% of the empty space at CPY on any day of the forecast period, while RB can consume at most 40%. On the final pass, any unused space at CPY is available to both RA and RB. Empty space at CPC is available to RA and RB on all passes because CPC is not key.

"Trim" (Max Release Variation) and Empty Space: When considering empty space at a control point, the method computes an upper bound on the flood control release for each timestep in the forecast period. The upper bound is the maximum amount that can be used as the first ordinate in a stepped-down release hydrograph from the timestep under consideration through the end of the forecast period, with the full reservoir's Maximum Release Variation (so-called "Trim" value) defining the magnitude of the reduction in this hydrograph between timesteps, and when this hydrograph is routed to the control point, all the resulting flow for the rest of the forecast period will fit in the empty space available at the control point. This stepped-down hydrograph is computed without regard to the goal release volume; it merely results in an upper bound on the release at each timestep. (This is a heuristic method of smoothing the release schedules. There is no guarantee that the second and succeeding timesteps of the stepped-down hydrograph will approximate the release hydrograph ultimately resulting from the flood control operations.)

Example: We are proposing a release from RB. For the purpose of this example, let us assume that below CPX, no other constraints reduce a release, that is, this example considers only the effects of CPB and CPX. RB has volume of 150 units to release to bring it down to the goal balance level for the pass. RB has a Max Release Variation of 10 flow units per timestep. Empty space at CPB for the forecast period is { 50, 50, 50, 50, 50 }. For the first forecast day, the empty space at RB would permit a stepped-down release hydrograph of {50, 40, 30, 20, 10}, so it proposes a release of 50 units. For the second day, the remaining empty space at CPB would then be { 0, 50, 50, 50, 50 }, so it proposes a release of 50 units again. Now the proposed release schedule is { 50,50, 0, 0, 0 } and the proposed empty space at CPB is {0,0,50,50,50}. Thus, CPB will impose the same upper bound on each day's release from RB.

Example: Continuing the above example, we now consider the empty space at CPX, which is { 50, 60, 40, 50, 50 }. For the first day, the stepped-down hydrograph would be 55, 45, 35, 25, 15}, which, routed, yields {27.5,50,40,30,20}, limited by day 3. Since the release is already constrained to 50 units on each day, CPX does not impose any constraints on the RB release schedule for the first day. Assuming the release from RB for the first day is indeed 50 units (after considering all control points and all other constraints), arriving at CPX as {25,25,0,0,0}, on the second day the resulting free space at CPX would be {25, 35, 40, 50, 50}. This empty space dictates a stepped-down hydrograph for day 2 of { 0, 45,35,25,15}, resulting in an arrival hydrograph of { 0, 22.5, 40, 30, 20 }, restricted by day 3. Thus, the upper bound of 45 is imposed on the release for day 2. Assuming that holds, we route it to CPX, and it arrives as {0,22.5,22.5,0,0}. This leaves empty space for day 3 at { 25,12.5,17.5,50,50}. This empty space dictates a stepped-down hydrograph for day 3 of {0,0,35,25,15}, which, routed to CPX gives an arrival hydrograph of {0,0,17.5,30,20}, being restricted by day 3. Thus, the upper bound of 35 is imposed on day 3, which we assume remains the release for day 3. Routed to CPX, it arrives as {0,0,17.5,17.5,0}, which reduces the empty space to {25,12.5,0,32.5,50}. The day 4 hydrograph is then { 0,0,0,55,45}, routed to CPX arriving as 0,0,0,27.5,50}, being restricted by day 5. Thus, the day 4 upper bound is 55, and if it ends up being released, it routes to CPX as {0,0,0,,27.5,27.5}, leaving empty space of {25,12.5,0,5,22.5}. This empty space dictates a day 5 hydrograph of {0,0,0,0,45}, which gives the upper bound of 45 for day 5. Thus control point CPX imposes limits of {55, 45,35,55,45}, and the release schedule is {50,45,35,55,45} .

While the above consideration of empty space guarantees that no release will cause flooding downstream, given the routing information and the forecast period limitations at hand, the following circumstances could cause flooding downstream at simulation time:

- the routing coefficients used by the flood control algorithm do not closely approximate the routing used during simulation, or
- the forecast period is considerably shorter than the effect of the routed releases at simulation time, or
- the model does not contain routing coefficients for control points reached within the forecast period by releases from reservoirs above, or
- there is a Flooding Exception relationship between a control point and a reservoir (see the Flooding Exception category on a control point for details).

Constraints of the Full Reservoir

The proposed release at any timestep in the forecast period will not exceed the release at the prior timestep plus the full reservoir's Allowable Rising Release Change. When computing this constraint on the release, surcharge and flood control minimum releases are considered.

Example: RA's Allowable Rising Release Change (ARRC) is 20. Computing RA's optimal release for the first day yields a release of 100. The total release (surcharge release plus flood control release) made on the prior simulation timestep was 75, so RA will not release more than 95 today (surcharge and flood control releases combined). After applying all other constraints, we end up with a release for today of 55. That means that the second forecast day's release will not exceed 75, and so on throughout the forecast period.

The algorithm tries to assure, but does not guarantee, that the proposed releases in the forecast period do not drop by more than the full reservoir's Allowable Falling Release Change. This constraint is applied by computing a stepped-down hydrograph that releases the goal volume for the timestep (based on the difference in storage between the forecast operating level at the end of the balance period and the balance level of the pass) over the remaining timesteps in the forecast period, with a reduction at each timestep of exactly the Allowable Falling Release Change within the forecast period (the last ordinate may exceed the Allowable Falling Release Change). The first ordinate in such a hydrograph is used as an upper bound on the release for the timestep under consideration. Of course, a release at some timestep may be forced to zero by sudden inflows at a downstream control point, and this can cause the Allowable Falling Release Change constraint to be untenable for that timestep. The upper bound computed on the first timestep (which uses the goal volume that is the entire flood pool's contents above the balance level of the pass) is applied to every timestep of the forecast period. Thus, the first timestep of the forecast period has one such upper bound applied, while the rest of the timesteps have two such upper bounds applied (one for the whole volume to release, and one for the remaining volume to release at that timestep).

Example: RA's Allowable Falling Release Change (AFRC) value is 10 flow units per timestep. RA is forecast to have150 units of flood pool volume at the end of the balance period, so 150 units is the release goal on a pass at balance level 5.0. The stepped-down hydrograph computed for 150 units volume for the first timestep is {50, 40, 30, 20, 10}, which releases all the volume in the forecast period. Thus, 50 is an upper bound on the first day's release. Having ultimately released 40 on the first day, due to space restrictions downstream, we compute a new hydrograph for the second day, to release 110 units in the 4 remaining days. This hydrograph is {42.5, 32.5, 22.5, 12.5}. (Note that the last day's ordinate may exceed AFRC, because our goal is first to release all the water within the forecast period, knowing that forecasts far out in the forecast period are dubious in any case.) The second hydrograph imposes an upper bound of 42.5, in addition to the upper bound of 50 on the release for the second day. Assuming we could ultimately release 41 units on day 2, we have for the third day 110-41=69 units to release. We compute a hydrograph for the third day: {33, 23, 13}. Thus the third day's release is subjected to the upper bound of 50 and the upper bound of 33. If a downstream control point imposes a limit of 0 on the third and fourth day releases, we will still have 69 units to release on day 5. This will give us a stepped-down hydrograph of {69}, and the upper bounds of 69 and 50 will apply to day 5. Thus, the upper bounds for each of the days for this constraint will be: {50,42.5,33,39.5,50}.

The proposed release at the simulation timestep may not exceed the spillway capacity, as determined by the reservoir's utility method getMaxOutGivenIn(). (This constraint is not applied to forecast timesteps after the first, that is, after the current simulation timestep.) If part of an upstream reservoir's proposed release is to flow through a downstream tandem reservoir (results in a so-called "through release"), the downstream tandem's spillway constraint is applied to the amount of the through release arriving on the simulation day plus other inflows so-far computed, along with other releases so-far computed for the tandem. Thus, the spillway constraint on the tandem may restrict flood control releases from upstream reservoirs.

Example: RA can release 100 units, and RC cannot store any water (having stored plenty from RB). Routing 100 units from RA to RC, we get {50,50}, and we then apply the spillway constraint at RC, considering the 50 arriving as being in addition to any other inflows at RC, and considering any other proposed releases from RC. If the release from RA, upon flowing through RC, could exceed the spillway capacity of RC, RA's release will be reduced accordingly.

The proposed release at a timestep may not dip into the full reservoir's conservation pool on that forecast timestep. This constraint may be applied to every timestep in the forecast period, or only to the first timestep, based on the method selection in the Tandem Storage Management category (q.v.).

If the Key Control Point Max Release category selection is Max Release Applies, the maximum flood control release constraint applies: on all passes but the last, a reservoir's proposed release may not exceed the reservoir's Max Flood Control Release on any timestep. This value is computed by the Key Control Point Balancing method. If a reservoir is not subject to a key control point, this constraint has no effect.

Example: Key control point CPY assigns to RA a maximum release of 40 units, and to RB a maximum release of 30 units. If we have three passes, at levels 9.0, 5.0 and 5.0, on the first two passes, RA will release no more than 40 units on any day of the forecast period. RB will not release more than 30 units on any day of the forecast period. Thus even though RA may be forecast to be significantly higher than RB, it may not be able use all the downstream channel space otherwise available to it. Only on the final pass will these limits be lifted, and RA will be free to consume all remaining channel space, subject to its other constraints. Assuming that the key control point allocates these maxima in proportion to the fullness of the reservoirs, using the maxima may have the effect of allowing each reservoir to release something each timestep, rather than allocating all channel space to RA today, leaving RB "fuller" tomorrow, allocating all channel space to RB tomorrow, leaving RA "fuller" the next day, and so on (causing undesirable oscillating behavior). Of course, the effect depends on the key control point balancing method selected.

On the last pass, the release schedule at timesteps after the first are subjected to another upper bound, which is the larger of

- the value proposed on the prior pass for the timestep, and
- the value proposed on this pass for the prior timestep minus the trim value (Maximum Release Variation slot).

This constraint may be overridden by selecting the Entire Forecast Period method of the Last Pass Timesteps May Increase category.

Example: RB is forecast to be at level 10.0 and RC is forecast to be at level 5.0 at the end of the balance period. The key control point balancing method assigns level 7.0 to all three reservoirs. On the first pass, at balance level 7.0, RC takes on 100 units from RB, released as { 40, 30, 20, 10}. Since the forecast period is 5 days long, the arrival hydrograph for this release schedule from RB is {4, 19, 30, 25, 15}, but because we consider only that water that arrives within the balance period, we account for only the 53 units arriving on days 1-3, leaving 47 units that arrive after the end of the balance period. Thus, in future calculations based on the forecast storage in RC, 53 units of volume will be added to RC's forecast storage. RC will be free to release on day 1 an additional 4 units of volume, and on day 2 an additional 19 units of volume, and so on through the forecast period. If the Two-Reservoir Midpoint method is selected, the starting volume for RC will henceforth include 53 units of volume above its forecast volume at the end of the balance period.

Storing Water in Downstream Tandem Reservoirs

All or part of a proposed release may be stored in downstream "tandem" reservoirs if the downstream reservoir is not Surcharging (Surcharge Release equals zero or is not valid) at the current controller timestep. Such "tandem storage" is determined as follows:

- A downstream reservoir will store water to increase its forecast operating level to the smaller of the pass's balance level and the tandem reservoir's Target Balance Level (assigned by a controlling key control point). If a reservoir is not subject to any key control point and thus not given a balance level, the Top of Conservation Pool value will be used for the assigned balance level and an error will be issued. This error will abort the run.

- A downstream reservoir may store additional water if the Two-Reservoir Midpoint method is selected in the Tandem Balancing category HERE (Section 7.1.9).

Example: There are 4 passes at balance levels 9.0, 8.7, 5.0 and 5.0. RC, subject to more than one key control point (not shown in the example subbasin), has been assigned a balance level of 8.7. On the first pass (9.0), RA may store water in RC to bring RC up to the larger of 9.0 (balance level of the pass) and 8.7 (balance level assigned RC). On the second pass, RC will accept water to bring it no higher than 8.7 again. On the last two passes, it will accept no water, unless it is forecast to be in the conservation pool at the end of the balance period, in which case, it will accept water to fill the conservation pool. If the Two-Reservoir Midpoint method is selected, RC will, in addition, agree to accept more water. Let us assume, as in a prior example, that RA is forecast to be at 10.2, RB at 8.9, and RC at 9.5. RC will accept, from RA, that volume of water that, if instantaneously moved from RA to RC, would put RC and RA a the same operating level. That level will be somewhere between 10.2 and 9.5. RC will not initially accept any water from RB because RB is forecast to be lower than RC. If, on the first pass, RC is able to release enough water to bring it below the ending operating level of RA at the conclusion of the pass, RC will accept water from RB.

When the Key Control Point Space Use method is Key Control Point Balancing Share, and water is moved from a full reservoir to a tandem, “ownership” of key control point shares of the empty space are transferred from the full reservoir to the tandem in amounts equal to the resulting inflow at the tandem for each timestep of the forecast period. This allows the tandem to release the water through the key control point later, all else permitting. Transfer of such “ownership” applies only to water to be stored in the flood pool (not to water to be stored in the conservation pool, as it will not be drained by the flood control algorithm). This transfer of ownership is not necessary and is not computed when the Pass Behavior selection is Undo and Recompute Max Release.

Example: This example applies only when Pass Behavior is Compute Compute Additional Release. On the first pass, RA releases 100 units of volume to be stored in RC, as follows: {50,30,20,0,0}. The key control point, CPY, has allocated to RA {30, 30, 30, 30, 30}, to RB {10, 10, 10, 10, 10} and to RC {20, 20, 20,2 0,20}. Having stored 100 units of RA’s water in RC, on a later pass we might be able to release some or all of that 100 units from RC, but because that water originated at RA, we transfer its ownership to RC. Routing the release hydrograph to CPY, we get {12,5, 32.5, 32.5, 17.5, 5}, so we transfer such units of water in CPY’s accounts. Now CPY has allocated to RA: {17.5, 0, 0, 12.5, 25}, and to RC: {32.5, 50, 50, 37.5, 25}.

The quantity of water scheduled for tandem storage and accounted for in the tandem reservoirs depends on the method selection in the Tandem Storage Management category. For details of the accounting, see the method descriptions, below.

Note: Please also read the notes regarding the flood control methods in general, above.

7.1.4 Balance Level Determination

This category appears only if the Operating Level Balancing method is selected in the Flood Control category. Its purpose is to determine the set of balance levels that will drive the Operating Level Balancing flood control method.

7.1.4.1 Key Control Point Balance Levels (default)

The default way to drive the Operating Level Balancing method is to collect the key control points' balance levels (Balance Level) computed by the Operating Level Balancing method of the Key Control Point Balancing category, and to use this set of balance levels to determine the number of passes to run and the operating levels to use on each pass.

Slots Specific to this Method

☞ **Balance Levels Used**

Type: Agg Series Slot
Units: NONE
Description: The set of balance levels taken from the key control points for driving the algorithm.
Information: This slot is not visible on the open object dialog, it is only visible in the Special Results USACE methods tab of the Model Run Analysis (on the subbasin). It is for observing the operation of the algorithm during or after a run.
I/O: Output.
Links: May not be linked.

7.1.4.2 Iteration Balance Levels

The user may drive the algorithm by defining a set of balance levels that will be used for the passes of the flood control algorithm. The more levels given, the more passes will be run, at a computational cost. The fewer levels given, the coarser will be the balancing. This selection may make sense if the allowable rising or falling release change represents the release of enough water to change reservoir's storage by an amount that is a significant portion of an operating level.

Slots Specific to this Method

☞ **Iteration Balance Levels Slot**

Type: Table Slot Nx1
Units: NONE
Description: The set of balance levels to drive the algorithm, independent of the activity of the key control points.
I/O: Required Input.
Links: May not be linked.

7.1.4.3 Reservoir Operating Levels

This method collects the reservoirs' forecast operating levels (at the end of the balance period). The highest reservoir operating level is excluded, and added to the set are the key control point balance levels (Balance Level) computed by the Operating Level Balancing method of the Key Control Point Balancing category (this is the set of balance levels used with the default selection). This option is useful when there are no key control points. It attempts to bring the highest reservoirs down to the levels of lower reservoirs before it processes the lower reservoirs.

SLOTS SPECIFIC TO THIS METHOD

☞ **BALANCE LEVELS USED**

Type: Agg Series Slot
Units: NONE
Description: The set of balance levels taken from the key control points for driving the algorithm.
Information: This slot is not visible on the open object dialog, it is only visible in the Special Results USACE methods tab of the Model Run Analysis (on the subbasin) It is for observing the operation of the algorithm during or after a run.
I/O: Output.
Links: May not be linked.

7.1.5 Pass Behavior

This category appears only if the Operating Level Balancing method is selected in the Flood Control category.

The flood control method operates in one of two ways: on each pass it computes an additional release for each reservoir for the pass's balance level, or it undoes any prior pass's release schedule and recomputes from scratch, given the new balance level.

7.1.5.1 Undo and Recompute Max Release (default)

For a given full reservoir FR, on a pass at balance level BL, pretend FR's release schedule for the prior balance level did not happen. Thus, at each downstream control point, add back in the empty space consumed by FR's prior release schedule, and at each tandem reservoir, subtract out the tandem storage scheduled on the prior pass. Then recompute the best release schedule for the full reservoir, as before, but this computation respects a new set of space reservations and tandem storages from other reservoirs, as computed on this and prior passes.

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

7.1.5.2 Compute Additional Release

Whatever could be release from a full reservoir FR on any pass can still be released on the next pass, with the exception of the final pass, which is recomputed so that additional empty space will be used as early as possible in the release schedule.

This method can give different results than recomputing the release, due to interactions with other method selections and the function of the “trim” (Max Release Variation slot) value in the empty space computations at control points. See **TRIM (MAX RELEASE VARIATION) AND EMPTY SPACE** above.

The main benefit of this method is to reduce complexity and thus runs may be faster. No performance comparisons have been made as of this writing.

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

7.1.6 Key Control Point Max Release

This category appears only if the Operating Level Balancing method is selected in the Flood Control category. This method category and its methods exist for convenience of algorithm development, and may not be available in the future.

Each reservoir that is subject to a key control point may be assigned a Max Flood Control Release value by the key control point balancing method. This method determines whether the flood control uses that value in its computations. (If the value is not a valid, it is ignored, in any case.)

The Max Flood Control Release is a value determined by the Key Control Point Balancing method. It is applied on all but the final pass of the algorithm. See the discussion and example in the subsection **CONSTRAINTS OF THE FULL RESERVOIR**, above.

7.1.6.1 Max Release Applies (default)

The Max Flood Control Release is applied as an upper bound on each timestep of the forecast period, on all passes but the last.

The effect of applying this maximum is mixed. On one hand, it tends to allow each subject reservoir to release something, even if at the expense of a higher-priority reservoir. This may help to smooth the release schedules for reservoirs, to avoid oscillating behavior, in which on one day, RA releases but RB does not, and on the following day RB releases, while RA does not, then RA releases, then RB releases, and so on. On the other hand, it may subvert the principle of giving priority to the fuller reservoirs.

When a reservoir's maximum is zero, the flood control algorithm bypasses most of the work associated with projecting a release schedule during all but the last pass.

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

7.1.6.2 Max Release Ignored

The value is ignored. See the discussion of the Max Release Applies method (above).

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

7.1.7 Key Control Point Space Use

This category appears only if the Operating Level Balancing method is selected in the Flood Control category. This method category and its methods exist for convenience of algorithm development, and may not be available in the future.

The allocation of space by the key control point balancing method may be respected or ignored, for purposes of experimentation.

7.1.7.1 Key Control Point Balancing Share (default)

On all passes but the last, allow subject reservoirs to use only as much space at key control points as has been allocated to them. A reservoir that is not a subject of a key control point gets no share of that key control point's empty space on any pass but the last. On the last pass, all space is available on a first-come, first-served basis.

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

7.1.7.2 First-Come, First-Served

Ignore the space allocations at the key control points. Every full reservoir has rights to space at key control points on a first-come, first-served basis.

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

7.1.8 Operating Level Mapping

This category appears only if the Operating Level Balancing method is selected in the Flood Control category.

Throughout the flood control computations, the Operating Level Balancing method converts operating levels to storages and vice versa. The mappings used for these conversions are defined in the Operating Level Table, which is a periodic slot, that is to say, the mapping changes with time. This method category exists to determine which timestep is used for the Operating Level Table lookups (interpolations).

Note: This method exists for the purpose of approximating SUPER results.

7.1.8.1 Simulation Timestep (default)

Use the current rules controller's simulation timestep for the periodic slot lookups and interpolations. When selected, the current simulation timestep is the date used in the (interpolated) table look-up, even when looking up such things as forecast storages at the end of the balance period. Thus, if a storage V translates to operating level 8.0 on the current simulation timestep, but to operating level 8.1 at the end of the balance period, and V is the storage at the end of the balance period, the operating level at the end of the balance period will be considered 8.0.

Slots Specific to this Method

☞ **None**

7.1.8.2 Forecast Timestep

Use forecast timestep in question for the periodic slot lookups and interpolations, when looking up storages and levels at that timestep. Thus, if a storage V translates to operating level 8.0 on the current simulation timestep, but to operating level 8.1 at the end of the balance period, and V is the storage at the end of the balance period, the operating level at the end of the balance period will be considered 8.1.

Slots Specific to this Method

☞ **None**

7.1.9 Tandem Balancing

This category appears only if the Operating Level Balancing method is selected in the Flood Control category.

There may be many ways to balance reservoirs that are in tandem (one downstream from another). This category is a place-holder for alternative methods for doing this.

7.1.9.1 None

If selected, no special action is taken to balance tandem reservoirs. Any downstream tandem may serve to store water released from above on any pass, but only up to the lower of the tandem's assigned balance level (Balance Level) and the balance level of the pass. Note, the downstream reservoir can only store water if there is zero Surcharge Release.

Slots Specific to this Method

☞ **None**

7.1.9.2 Two-Reservoir Midpoint (default)

If selected, a downstream tandem may store water as above (None) and possibly a little more: it will find the midpoint of the forecast operating levels of the two reservoirs, and store the amount of water that will both bring the upstream reservoir to that midpoint and bring the downstream reservoir to that midpoint. (Of course, later in the algorithm, such stored water may be drained.) Note, the downstream reservoir can only store water if there is zero Surcharge Release.

Starting storages for two-reservoir balancing are as follows:

- Upstream reservoir: the original forecast storage at the end of the balance period plus tandem storage scheduled here from upstream (through either the end of the forecast period or the end of the balance period, depending on the Tandem Storage Considered method selection, q.v.) minus the volume already scheduled to be stored in intervening reservoirs.

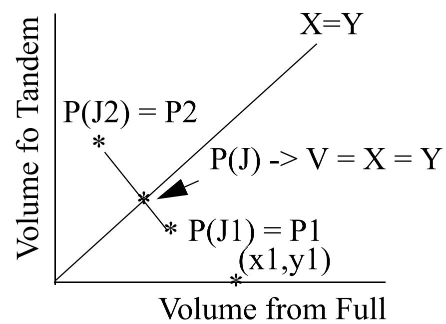
- Downstream tandem reservoir: the original forecast storage at the end of the balance period plus tandem storage scheduled here from other upstream reservoirs (through either the end of the forecast period or the end of the balance period, subject to the Tandem Storage Considered method selection, q.v.) minus the volume released on the first timestep by any proposed flood control release from a prior pass, with a floor of the storage equivalent to the lesser of the balance level of the pass and the assigned balance level. (This method is not invoked until the flood pool has already been filled to this floor level.)

The method is invoked at most once for each {upstream, tandem} pair in any pass of the algorithm, and then only after the tandem's conservation pool is filled and its flood pool is filled up to the smaller of the pass balance level and the assigned balance level of the tandem reservoir. Once invoked, this method computes a balance level B (between the balance levels for the starting storages) and volume V such that moving volume V from the upstream reservoir to the tandem will leave them both at level B. The return value of the method is the volume V, and the flood control algorithm is free to move up to V from the upstream to the downstream reservoir. By this means, and only by this means, will a downstream reservoir be filled above the balance level of a pass on that pass, or above its assigned balance level on any pass.

If the downstream reservoir's projected balance level is higher than that of the upstream reservoir (based on the starting volumes as described above), the return value will be 0.0 (no additional tandem storage).

The details of the algorithm for finding the volume V are as follows:

The method is looking for a point of intersection between two lines:



Line 1 is the line X=Y.

Line 2 is the line between two points P1 and P2.

Each point P(J) represents what it would take to bring the full reservoir down to level J and the tandem up to level J.

P(J) = (volume that the full reservoir contains above level J, volume that can be stored in the tandem to take it to level J).

The method finds the balance level J such that P(J) is on the line X=Y. This is done in two steps: find points P1 and P2 where P1 is in the lower half of the space defined by the line X=Y and P2 is in the upper half. Then the method finds the point of intersection between the line (P1,P2) and the line X=Y.

Step 1: Find points P1 and P2. Start with

x1=the volume that the full reservoir contains above the tandem's balance level, and

y1=0

This point (x1,y1) represents bringing the full reservoir to the tandem's level and storing nothing in the tandem. Clearly this point is in the region where x > y.

Now find a point P(J) for some J that falls in the upper half of the graph defined by X=Y, that is, where x <= y:

Start with J = tandem's balance level + 1.

P(J) = (volume that the full reservoir contains above level J, volume that can be stored in the tandem to take it to level J).

If x <= y, then P(J) falls in the upper half of the plane and P2 is P(J). If not, set P1 = P(J) and increment J by 1, and recompute P(J) until a P(J) is found that is in the upper half of the graph. This is P2.

Step 2: Now that we have a J such that x <= y, we find the point of itersection between P1 and P2. Compute the slope of the line between P1 = (x1,y1) and P2=(x2,y2):

$$slope = \frac{y2 - y1}{x2 - x1}$$

The volume V to release is:

$$V = \frac{y1 - x1 \times slope}{1 - slope}$$

Invoking this method on one pair of reservoirs changes the state of the downstream reservoir, and subsequent invocation with a different upstream reservoir may change it yet again. Repeated invocations over several passes tends to bring the collected upstream reservoirs in balance with the tandem, thus, to some extent, with each other.

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

7.1.10 Modify Inputs to Two-Reservoir Midpoint

This category appears only if the Two-Reservoir Midpoint method is selected in the Tandem Balancing category. This method category allows the user to modify the behavior of the Two-Reservoir Midpoint method to more closely mimic SUPER. The Two-Reservoir Midpoint method, as described above, takes into acccount the volume of tandem storage that has already been scheduled to be stored in the upstream reservoir by a reservoir upstream of it (this could have happened if the more upstream of the two were "fuller" and therefore processed first). The SUPER algorithm does not take this volume into account, so RiverWare offers this category so the user can select whether to mimic SUPER or to use the Two-Reservoir Midpoint method as described above.

The two methods in this category are described below.

Note: This category exists for the purpose of approximating SUPER results.

7.1.10.1 Omit Tandem Storage from Upstream Reservoir (default)

SUPER does not include tandem storage in the upstream reservoir for the purpose of computing the starting storage of the upstream reservoir. Tandem storage is considered in the downstream reservoir.

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

7.1.10.2 None

Include tandem storage in both reservoirs.

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

7.1.11 Tandem Storage Management

This category appears only if the Operating Level Balancing method is selected in the Flood Control category. This method category allows the user to control the accuracy of the tandem storage calculations used to propose flood control releases.

The SUPER algorithm treats tandem storage as blocks of water moved, without regard to the times they are released or arrive.

Note: This method exists for the purpose of approximating SUPER results.

7.1.11.1 Do Not Route Tandem Storage (default)

SUPER treats tandem storage as blocks of water moved instantaneously, without regard to the times they are released or arrive. This method allows RiverWare to more closely mimic the SUPER behavior. Consequences of not routing the water are:

- water that is scheduled for tandem storage cannot be released as soon as it arrives; it can be released only at the end of the release schedule, and
- such water is released at the end of the release schedule, even if it has not been released from above yet, or might all have arrived after the end of the forecast period, and
- such water is accounted for *in toto* when computing a Two-Reservoir Midpoint balance, or for computing space left in the flood pool for holding more tandem storage, regardless when it is released or might arrive, and
- since the algorithm does not know when this water arrives at a tandem reservoir, it does not check, after the first timestep, that a release does not dip into the conservation pool.

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

7.1.11.2 Route Tandem Storage

Water scheduled for tandem storage is routed, and its arrival times are known to the Operating Level Balancing method, so it can release the water as soon as possible. For the purpose of computing starting storages for the Two-Reservoir Midpoint, the method can be more accurate in its knowledge of the storages in the two reservoirs. Nevertheless, there are options for selecting the quantity of tandem storage water for which to account (see the Tandem Storage Considered category, below).

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

7.1.12 Tandem Storage Considered

This category appears only if the Route Tandem Storage method is selected in the Tandem Storage Management category. When water scheduled for tandem storage is routed, its arrival times are known.

The flood control algorithm balances reservoirs based on their projected storages at the end of the balance period. In a sense, the end of the balance period is the end of the time when there is some confidence in the forecasts. The question arises: if we release water today to store downstream, we know it will arrive, so do we care *when* it arrives? Should the fact that we released it be enough to consider it stored at its destination? This category lets us choose one of two ways of answering that question. Myriad other possibilities (such as considering only water released on the first timestep, or scheduling storage only on the first timestep or only during the balance period)   are not implemented.

7.1.12.1 Arrives in Forecast Period (default)

The flood control algorithm considers all the water that arrives at a tandem within the forecast period to contribute to the tandem's storage. This means that water released after the end of the balance period is considered also.

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

Example: RB is forecast to be at level 10.0 and RC is forecast to be at level 5.0 at the end of the balance period. The key control point balancing method assigns level 7.0 to all three reservoirs. On the first pass, at balance level 7.0, RC takes on 100 units from RB, released as { 40, 30, 20, 10}. Since the forecast period is 5 days long, the arrival hydrograph for this release schedule from RB is {4, 19, 30, 25, 15}, leaving 7 units that are routed off the end of the forecast period. Thus, in future calculations based on the forecast storage in RC, 93 units of volume will be added to RC's forecast storage. RC will be free to release on day 1 an additional 4 units of volume, and on day 2 an additional 19 units of volume, and so on. If the Two-Reservoir Midpoint method is selected, the starting volume for RC will henceforth include 93 units of volume above its forecast volume at the end of the balance period.

7.1.12.2 Arrives in Balance Period

The flood control algorithm considers only that water that arrives within the balance period to contribute to the tandem's storage. This means that water released within the balance period is the only water considered.

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

7.1.13 Priority Determination

This category appears only if the Operating Level Balancing method is selected in the Flood Control category.

Methods in this category determine the timing of the priority determination of reservoirs.

7.1.13.1 None (default)

This is not a valid method; an error will be posted if selected.

7.1.13.2 End of Prior Timestep, Fixed

The reservoirs in the subbasin are given priorities based on their "fullness" at the end of the prior simulation timestep, if in the flood pool at that time. If a reservoir is in the flood pool at the end of the prior timestep, its ending operating level is its priority. If not, its forecast operating level at the end of the balance period becomes its priority. These priorities are used for the entire duration of the flood control method, regardless how much water is scheduled for release from any reservoir in the subbasin during the method's computations.

When this method is selected, the set of "full" reservoirs on any pass is processed in the order of the predetermined priorities, above. If reservoirs are to be added to the set, the Reservoir Set category (q.v., below) selection must be Reservoirs Starting in or Entering Flood Pool.

SLOTS SPECIFIC TO THIS METHOD

☞ **FORECASTED OPERATING LEVEL**

Type: Agg Series Slot
Units: LENGTHS
Description: There is one column for each reservoir in the subbasin (this is recreated at run initialization). Each column stores the forecasted operating level; that is for

the current timestep, the operating level a the end of the balance period. Thus, if the balance period is 4. The value shown on this slot at timestep 1 is the forecasted operating level at timestep t+3 (current plus three timesteps). This operating level is from the first pass only.

Information: This slot is invisible but can be viewed in the computational subbasin's Special Results details on the Model Run Analysis tool **HERE (USACE_SWD.pdf, Section 5.5)**.

I/O: Output only

Links: Not Linkable

Example: RA ended yesterday's simulation at level 4.8, RB at 8.1, RC at 4.4. RA is forecast to be at level 5.6 at the end of the balance period, RB at 10.0, and RC at 4.9. The priorities are assigned as follows: RA: 5.6, RB 10.0, RC 4.4. On each pass, the reservoirs will be processed (given a chance to consume channel space) in this order: RB, RA. RC will not be processed, unless the Reservoir Set category selection is Reservoirs Starting in or Entering Flood Pool, in which case, it will be processed on the last pass if it takes on tandem storage to bring it into the flood pool.

7.1.13.3 End of Balance Period, Varies

Flood control considers only the reservoirs' forecast operating levels at the end of the balance period. Those priorities are recomputed at each pass of the algorithm, based on the latest forecast storages, which take into account proposed release schedules and proposed tandem storage. At the beginning of each pass, the entire subbasin's set of reservoirs is considered. The last pass is demand-driven, so that if reservoir A's new releases change the state of empty space in a way that might affect reservoir B, reservoir B is reprocessed.

Slots Specific to this Method

☞ Forecasted Operating Level

Type: Agg Series Slot

Units: Lengths

Description: There is one column for each reservoir in the subbasin (this is recreated at run initialization). Each column stores the forecasted operating level; that is for the current timestep, the operating level a the end of the balance period. Thus, if the balance period is 4. The value shown on this slot at timestep 1 is the forecasted operating level at timestep t+3 (current plus three timesteps). This operating level is from the first pass only.

Information: This slot is invisible but can be viewed in the computational subbasin's Special Results details on the Model Run Analysis tool **HERE (USACE_SWD.pdf, Section 5.5)**.

I/O: Output only

- Not Linkable

Example: RA ended yesterday's simulation at level 4.8, RB at 8.1, RC at 4.4. RA is forecast to be at level 5.6 at the end of the balance period, RB at 10.0, and RC at 4.9. The initial priorities are assigned as follows: RA: 5.6, RB 10.0, RC 4.9. On the first pass, the reservoirs will be processed (given a chance to consume channel space) in this order: RB, RA. As soon as RC takes on enough storage to put it above the balance level of a pass, it will be processed. Likewise, the other reservoirs will be re-prioritized on each pass, according to their projected levels that result from the projected releases on the prior pass.

7.1.14 Reservoir Set

This category appears only if the Priority Determination category selection is End of Prior Timestep, Fixed. This category allows a variation on that method.

Note: This method exists for the purpose of approximating SUPER results.

7.1.14.1 Reservoirs Starting in Flood Pool (default)

Reservoirs will not be added to the set of reservoir considered on any pass. If a reservoir starts in the conservation pool and takes on tandem storage, which puts it into the flood pool, it will be considered for making releases in this invocation of the Operating Level Balancing flood control method. Such water will not be considered for release until the next simulation timestep.

Slots Specific to this Method

- **None**

7.1.14.2 Reservoirs Starting in or Entering Flood Pool

If a reservoir starts in the conservation pool and takes on tandem storage, which puts it into the flood pool, it will be considered for making releases in this invocation of the Operating Level Balancing flood control method.

Slots Specific to this Method

- **None**

7.1.15 Smoothing Releases

This category appears only if the Operating Level Balancing method is selected in the Flood Control category. This method category and its methods exist for convenience of algorithm development, and may not be available in the future.

The Operating Level Balancing method applies heuristic schemes (Allowable Falling Release Change, Allowable Rising Release Change, and Maximum Release Variation) designed to smooth the reservoirs' release graphs, as described above, by limiting the change in release from one timestep to the next, or by assuming a stepped-down hydrograph with a given change at each timestep. The releases considered at each timestep may include: the so-far-computed flood control release, the surcharge release, the flood control minimum release and the "through release". "Through release" refers to water released from upstream reservoirs that flows through a downstream tandem reservoir. This water is the "through release" of the tandem. The through releases may be considered among, or excluded from, the set of releases used in the smoothing heuristics. Through releases are ultimately folded into the flood control releases, but at the time of the smoothing operations, they are not yet folded in.

Note: This method exists for the purpose of approximating SUPER results.

7.1.15.1 Consider Flood Control and Surcharge Only (default)

Do not consider through releases (or any other releases that might be added with future development).

When checking for empty space at a control point (applying Max Release Variation), do not consider surcharge or through releases if the method selection for Pass Behavior is Undo and Recompute Max Release.

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

7.1.15.2 Consider All

Consider all releases that have been computed for this reservoir at the time the smoothing operation is applied.

If the method selection for Pass Behavior is Undo and Recompute Max Release, this method selection has no effect on the smoothing operation using Max Release Variation (trim) when checking for empty space at control points. When running with Undo and Recompute Max Release, only flood control releases are considered for smoothing purposes at control points.

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

7.1.16 Last Pass Timesteps May Increase

This category appears only if the Operating Level Balancing method is selected in the Flood Control category. This method category and its methods exist for convenience of algorithm development, and may not be available in the future.

When the Operating Level Balancing method runs the final pass of the flood control algorithm, it may subject each timestep's proposed releases to the same constraints as prior passes (except the Key Control Point Max Release and Key Control Point Space Use never apply on the last pass), or it may add an additional constraint to timesteps after the first.

Note: This method exists for the purpose of approximating SUPER results.

7.1.16.1 Current Simulation Timestep Only (default)

Limit increases in releases after the first timestep of the schedule on the last pass. Releases on timesteps after the first are subject to one more constraint on the final pass: that they do not exceed the larger of

- the value proposed on the prior pass for the timestep, and
- the value proposed on this pass for the prior timestep minus the trim value (Maximum Release Variation slot).

This method allows some reservoirs to make larger releases than they might otherwise make, and reduces releases for some reservoirs on the current simulation timestep. Its effect is not predictable.

Slots Specific to this Method

☞ **None**

Example: If the Last Pass Timesteps May Increase selection is Current Simulation Timestep Only, this constraint is applied. After the first two of three passes, RA's proposed release schedule is {50, 70, 90, 30, 10}. Its Maximum Release Variation (trim) is 10, and its Allowable Rising Release Change is 20. Now we run the final pass, and, having eliminated the constraints imposed by the key control point CPY, we could now release {80, 100, 100, 100, 90}. This constraint says that we will release {80, 70, 90, 80, 70}. The first timestep is unconstrained here; the second is constrained by both (or either, since they are the same) the value proposed on the prior pass (70) and the value proposed on this pass, prior timestep (80) minus the trim value (10). The third day is constrained by the prior schedule. The fourth day is constrained by 90-10=80, which is larger than 30, so it is the upper bound, and the last day is also constrained by the prior timestep minus trim.

7.1.16.2 Entire Forecast Period

Let the last pass behave like all other passes, except the Key Control Point Max Release and Key Control Point Space Use never apply on the last pass.

Slots Specific to this Method

☞ **NONE**

7.1.17 Water Rights Allocation

The selected method in this category is executed by the RPL function SolveWaterRights(), described **HERE (RPLPredefinedFunctions.pdfRPLPredefinedFunctions.pdf, Section 175)**.

7.1.17.1 None

There are no slots or calculations associated with this method.

7.1.17.2 Prior Appropriation

This method, if selected, is executed by the RPL function SolveWaterRights(), described **HERE (RPLPredefinedFunctions.pdfRPLPredefinedFunctions.pdf, Section 175)**. For more information on the Water Rights Allocation solver, click **HERE (Accounting.pdf, Section 5)**. The method allocates water from a specified “allocatable flow” supply chain to prioritized water accounts in the specified subbasin, meeting the “first in time, first in right” requirement.

SLOTS SPECIFIC TO THIS METHOD

☞ **TOP OF CONSERVATION POOL**

Type: Scalar Slot
Units: NONE
Description: The Operating Level that represents the top of the conservation pool
Information: This slot appears on the subbasin only for the convenience of propagating the value to all reservoirs in the subbasin (propagation is optional). Reservoirs that hold any storage rights accounts must have the Top of Conservation Pool slot defined. (See the documentation for the Reservoir Object, Operating Levels category.)
I/O: Input only
Links: Not linkable

☞ **SHORTAGE TOLERANCE**

Type: Scalar Slot
Units: FLOW
Description: Any shortage less than this amount will be considered zero by this water rights allocation method. Choosing a non-zero value for this slot will reduce the amount of processing spent on very small allocation differences (e.g., in making cutbacks).
Information:
I/O: Input only
Links: Not linkable

☞ **Lag Distance**

Type: Scalar Slot
Units: NONE
Description: Maximum lag in number of timesteps from any headwater passthrough account in the subbasin to the downstream-most passthrough account in the subbasin. This slot keeps track of passthrough account lags only (*not* return flow lags).
Information: This value is computed when the subbasin is verified (either through the GUI due to user actions, or at the beginning of a run), and is used to compute the local timestep offsets of passthrough accounts in the subbasin.
I/O: May not be input.
Links: Not linkable

☞ **Minimum Appropriation**

Type: Scalar Slot
Units: FLOW
Description: Any potential appropriation less than this amount will be reduced to zero by this water rights allocation method. Choosing a non-zero value for this slot will reduce the processing spent on making very small allocations. NOTE: RiverWare will use a minimum of 1e-12 cms for this value. This is not a default; it is the smallest functional value that RiverWare will use, even if you choose a smaller value (e.g., zero) for this slot.
Information:
I/O: Input only
Type: Not linkable

7.1.18 Account Equal Priority Allocation

The selected method in this category is executed by the RPL function SolveWaterRights(), described HERE (RPLPredefinedFunctions.pdfRPLPredefinedFunctions.pdf, Section 175). The method specifies how allocation of water is calculated for accounts that have equal priority dates.

7.1.18.1 None

No method is selected for handling equal priority accounts during the SolveWaterRights RPL function and an error will be generated if there are accounts with equal priority dates in the computational subbasin when the RPL function is executed.

7.1.18.2 Share Proportionally with Limits

This method, if selected, is executed under the RPL function SolveWaterRights(), described HERE (RPLPredefinedFunctions.pdfRPLPredefinedFunctions.pdf, Section 175). When two or more accounts are encountered with an equal priority date, this method iteratively shares available water based on the proportion of cumulative water right to available flow, limited by the minimum downstream proportion. For a more detailed explanation of this calculation, see information on the Water Rights Allocation solver **HERE (Accounting.pdf)**.

7.1.19 Account Initial Request

These methods disaggregate **Annual Request** (user input) slot on Diversion Accounts from annual timestep to daily or monthly timesteps on a set of accounts in the subbasin (typically used for water rights appropriations; for more information, click **HERE (Accounting.pdf)**). The result is written into the **Initial Request** slot on the account. Initial Request is used by the Water Rights solver. You may specify the multipliers used to disaggregate the demands in terms of periodic or series data. In either case, RiverWare selects the proper periodic or series slots to use based on the timestep size from the run controller. Two timestep sizes are supported: **day** and **month**.

The method operates only on diversion accounts on objects in this subbasin on which the "Disaggregated by Subbasin" method (**HERE (Accounting.pdf)** is selected in the Initial Request category.

The selected method is executed at the beginning of the run if this subbasin is enabled. It disaggregates the annual timesteps into timesteps of the Initial Request slots on the accounts for the range of timesteps that covers both the run period and the accounting period.

Conceptually, the coefficients have units of 1/sec, so the Initial Request value for a given timestep is a flow, which is stored in internal units (cms).

flow = (Annual Demand in m3) * coefficient / seconds-in-the-given-timestep.

If not all accounts in the subbasin use the same coefficients for disaggregation, alternative subbasins can be defined for each set of accounts that share coefficients.

7.1.19.1 None

There are no slots or calculations associated with this method. Select this method if you are not interested in disaggregating annual demands.

7.1.19.2 Periodic Coefficients

This method uses a periodic slot to define the coefficients.

SLOTS SPECIFIC TO THIS METHOD

☞ **DAILY DEMAND COEFFICIENTS**

Type: Periodic Slot
Units: NONE
Description: Multipliers to be applied to an account's Annual Request slot value to yield an Initial Request slot value in a non-leap year when the run timestep is a day.
Links: Not linkable.

☞ **DAILY LEAP YEAR DEMAND COEFFICIENTS**

Type: Periodic Slot
Units: NONE
Description: Multipliers to be applied to an account's Annual Request slot value to yield an Initial Request slot value in a leap year when the run timestep is a day.
Links: Not linkable.

☞ **MONTHLY DEMAND COEFFICIENTS**

Type: Periodic Slot
Units: NONE
Description: Multipliers to be applied to an account's Annual Request slot value to yield an Initial Request slot value in a non-leap year when the run timestep is a month.
Links: Not linkable.

☞ **MONTHLY LEAP YEAR DEMAND COEFFICIENTS**

Type: Periodic Slot
Units: NONE
Description: Multipliers to be applied to an account's Annual Request slot value to yield an Initial Request slot value in a leap year when the run timestep is a month.
Information: Not linkable.

7.1.19.3 Series Coefficients

This method uses a series slot to define the coefficients.

SLOTS SPECIFIC TO THIS METHOD

☞ **DAILY DEMAND COEFFICIENT SERIES**

Type: Series Slot
Units: NONE
Description: Multipliers to be applied to an account's Annual Request slot value to yield an Initial Request slot value when the run timestep is a day.
Links: Not linkable.

☞ **MONTHLY DEMAND COEFFICIENT SERIES**

Type: Series Slot
Units: NONE
Description: Multipliers to be applied to an account's Annual Request slot value to yield an Initial Request slot value when the run timestep is a month.
Links: Not linkable.

7.1.20 Local Inflow Spatial Disaggregation

The Local Inflow Spatial Disaggregation category contains three methods: the default, no action method, None, the WAM Precipitation Curve Number method, and the Drainage Area method. Both the WAM Precipitation Curve Number method and the Drainage method take known gage flows and estimate the flow at intervening or upstream Ungaged Control Points. The WAM Precipitation Curve Number method calculates an equivalent precipitation, using an empirical equation to relate watershed parameters, gage flows and precipitation. The Drainage Area method calculates a ratio between incremental gage drainage area and the ungaged drainage area to distribute flows. Spatial disaggregation is calculated at the start of the run for all enabled subbasins.

The three categories of disaggregation methods are executed in the following order: (1) Local Inflow Spatial Disaggregation, (2) Local Inflow Temporal Disaggregation, and (3) Incremental Local Inflows. The last category, Incremental Local Inflows, may include forecasting.

The computational subbasin executes the disaggregation methods at the beginning of a run and subsequently sets the relevant disaggregated local inflow slots as inputs on control points and reservoirs. In practice, users may run the model once to perform the disaggregation, then disable the subbasins related to disaggregation and save the model. When this method is used, future model runs will not execute the disaggregation methods and users will avoid the computational expense of disaggregating the local inflows at the beginning of each run. Should users wish to re-calculate their disaggregation methods, they should re-enable the subbasins and start a model run.

While the actual computation of the disaggregation occurs via the three methods on the computational subbasin, the data for the disaggregation is stored in method dependent slots on control points and reservoirs.

7.1.20.1 None

This method is the default for the Local Inflow Spatial Disaggregation category and should be selected when spatial disaggregation of local inflows is not desired.

There are no slots specifically associated with this method.

7.1.20.2 WAM Precipitation Curve Number

This method computes the Distributed Flow values for all Ungaged Control Points in the subbasin given the known Distributed Flow values of the downstream Gage Control Point, upstream Gage Control Points (if any), and excluded Gage Control Points (if any). There are no slots specifically associated with this method on the subbasin. The relevant data is held in method dependent slots on the control points and is accessed by the computational subbasin. See the Local Inflow Spatial Disaggregation on Subbasin method on the Control Point, HERE (Section 9.1.3.2), for details of these slots.

The WAM Precipitation Curve Number method disaggregates using methods found in the Water Availability Model (WAM) developed by Dr. Ralph Wurbs at Texas A&M University. The WAM Precipitation Curve Number method is executed on the computational subbasin at the beginning of run if the subbasin is enabled. The method calculates the Distributed Flow at each ungaged control point using the NRCS curve number, mean precipitation, and drainage area of both the gaged and ungaged control points.

The downstream Gage Control Point calculates the gain in the basin, using the difference in Distributed Flow between the downstream gage control point and its upstream gage control point(s) as defined in the downstream gage. The effect of excluded gages is defined below.

The downstream Gage Control Point calculates the area of the basin, subtracting the area of all upstream gages from the area provided in the downstream gage. The upstream gage curve number and upstream gage mean precipitation are scaled by area and subtracted from the downstream gage curve number and mean precipitation, also scaled by area. The resulting curve number and mean precipitation are scaled by 1/area to calculate the updated curve number and mean precipitation.

Excluded gages may be defined in the Ungaged Control Point, to change the gain calculation for its downstream gage control point. For most basins, the Excluded gages list is not needed. Each Ungaged Control Point maintains its own list of excluded gages. When a gage is placed in the list, the downstream gage ignores that excluded gage when solving the gain, curve number, and mean precipitation at the downstream control point.

The algorithm is described in the steps below:

1. All of the Ungaged Control Points in the computational subbasin are visited to perform the spatial disaggregation of flows to Ungaged Control Points. For each Ungaged Control Point to be calculated, the Downstream Gage is visited, and the incremental flow for that gage is calculated. Distributed Flow from upstream gages is subtracted from the Distributed Flow in the currently computed gage. Any upstream gages that are not tributary to the ungaged point may be left in the gage (not subtracted) using Excluded Gages list. This list may be different for each Ungaged Control Point, based on its location in the subbasin. The resulting set of incremental flows is only used for calculations pertaining to the current Ungaged Control Point.

2. The drainage areas are checked. An error occurs if the total drainage area of the Upstream Gage Control Points are equal or larger than the drainage area of the Downstream Gage Control Point or the Ungaged Control Point.

3. Incremental drainage area at the Downstream Gage Control Point is calculated, subtracting the sum of the drainage area for all Upstream Gage Control Points, except for those specified in the Excluded Gage Control Points list.

4. The incremental curve number at the Downstream Gage Control Point is calculated. The curve number is area weighted, subtracting the area weighted curve numbers from Upstream Gage Control Points, except for those specified in the Excluded Gage Control Points list.

5. The incremental mean precipitation at the downstream gage is calculated. The mean precipitation is area weighted, subtracting the area weighted mean precipitation from Upstream Gage Control Points, except for those specified in the Excluded Gage Control Points list.

6. Incremental drainage area at the Ungaged Control Point is calculated, subtracting the sum of the Drainage Area for all Upstream Gage Control Points.

7. The incremental curve number at the Ungaged Control Point is calculated. The Curve Number is area weighted, subtracting the area weighted curve numbers from Upstream Gage Control Points.

8. The incremental mean precipitation at the Ungaged Control Point is calculated. The mean precipitation is area weighted, subtracting the area weighted mean precipitation from Upstream Gage Control Points.

9. The incremental runoff (Q) at the Downstream Gage Control Point is computed by dividing the monthly incremental inflow at the Downstream Gage Control Point by its incremental drainage area.

10. The precipitation depth (P) at the Downstream Gage Control Point is calculated through an iterative solution (bisection) given the runoff computed in the previous step and the value of S as follows.

$$Q = \frac{(P - 0.2S)^2}{P + 0.8S} \qquad P \geq 0.2S$$

$$Q = 0 \qquad P < 0.2S$$

$$S = \frac{1000}{CN} - 10$$

11. If the Downstream Gage Control Point runoff (Q) is less than or equal to zero, the precipitation depth (P) at the Downstream Gage Control Point is 0.2 times the value of S.

12. The precipitation depth at the Ungaged Control Point is computed by adjusting the precipitation depth at the Downstream Gage Control Point by the ratio of the mean precipitation depth (M) at the ungaged and gaged control points.

$$P_{ungaged} = P_{gaged}\left(\frac{M_{ungaged}}{M_{gaged}}\right)$$

13. In order to avoid a divide by zero error and the better match reality, the Downstream Gage Control Point's mean precipitation depth must be greater than zero. For consistency, the Ungaged Control Point's mean precipitation depth must also be greater than zero.

14. The runoff at the Ungaged Control Point is then computed using curve number equation using $P_{ungaged}$ and $S_{ungaged}$ for the Ungaged Control Point.

15. The computed value for the runoff is then converted to incremental streamflow. Incremental streamflow is added to the Upstream Gage Control Point streamflow to calculate an initial estimate of streamflow at the Ungaged Control Point.

16. The initial estimate is compared to the downstream gage flow, and the minimum is selected. The results at the upstream gage point never exceed the downstream gage flows.

17. Set the results to the Distributed Flow slot as inputs.

The iterative solution for Precipitation in Step 10 is solved using the bisection method. The bisection routine is an incremental search method in which the function is reevaluated at the midpoint of the interval between the values of the previous two guesses to determine on which side of the midpoint the root lies. Depending on which side of the midpoint the root lies, the midpoint becomes either the upper or lower bound to the search interval. The interval is again divided in half and the function is reevaluated at the midpoint. This procedure is repeated until the solution is obtained. The bisection method typically uses the maximum and minimum values for the unknown variable and finds the midpoint of these to use as the seed value to start the iteration. While the minimum value of Precipitation is zero, the maximum value is not clearly defined at the start of step 10, thus a seed value of P = Mean Precipitation will be used to start the bisection routine. If the right hand side (RHS) of the equation in step 10 is smaller than the runoff (the left hand side (LHS) of the equation) the value of P will become twice the previous value for P (i.e., twice the Mean Precipitation). This will be repeated until the RHS is larger than the LHS, indicating that an upper bound for P has been determined. From here the bisection method continues in the usual fashion. If the seed value of P results in the RHS of the equation in step 10 being larger than LHS, the value of P will become 0.5 times Mean Precipitation (i.e., the midpoint between the minimum value, 0, and the initial value, Mean Precipitation). From here the bisection method proceeds in the typical fashion. The bisection method will continue until two successive solutions are within convergence of each other or until maximum iterations is reached. Convergence is defined as the convergence criteria set on the gage control point's Distributed Flow slot.

Note that the empirical equations in this method use units of acres for drainage area, acre-feet per month for flow, inches per month for runoff and inches for precipitation. Curve Number is unitless. Users may enter data in any units and the values will be internally converted for the computation.

The method will set the non-gage control points' Distributed Flow slots as input to avoid these values being cleared in subsequent model runs. After the WAM Precipitation Curve Number is completed, all control points will contain a value in the Distributed Flow slot for all timesteps in the slot. After the WAM Precipitation Curve Number method is completed, data is made available for the next stage of the disaggregation or forecast depending on which is the next selected method. Note that all steps of the disaggregation and forecast will be taken in sequence: the local inflow spatial disaggregation (if any) occurs first, the local inflow temporal disaggregation (if any) occurs next, and then the calculation of incrementals (if any). If a non-none Local Inflow Temporal Disaggregation method is the next selected method, the Distributed Flow values will be used in the temporal disaggregation calculations. If the Compute Incremental Local Inflows method is the next selected method, Distributed Flow values are copied into the Cumulative Local Inflow slot for the computation of incremental flows, but only if the timestep sizes of the two slots match. If no other disaggregation or forecast method is selected, the Disaggregated Flow slot is copied into the Local Inflow slot. This copying of slots occurs during beginning of run.

The default timestep size for the Distributed Flow and Mean Precipitation slots is months. This can, however, be changed by the user by configuring the timeseries and is independent from the model timestep size. It is up to the user to make sure the timestep size and associated data are appropriate.

Model Setup

The computational subbasin must be set up for each subbasin in which flows will be spatially disaggregated. On the computational subbasin, select WAM Precipitation Curve Number method from the Local Inflow Spatial Disaggregation category. Append the relevant control points to the subbasin. Each spatial disaggregation subbasin should include at least one Ungaged Control Point, where flow will be determined using the Gage Control Point(s) as a reference. Each Ungaged Control Point should be included in only one subbasin.

For each control point in the computational subbasin, the WAM Precipitation Curve Number method must be selected from the Local Inflow Spatial Disaggregation category. Selecting the WAM Precipitation Curve Number method enables the Gage Control Point category. Select the Gage Control Point method from the Gage Control Point category for the each gage control point in the basin. Note that upstream control points do not need to be a member of this subbasin, but they need to be a member of at least one subbasin. All other control points should have None selected in the Gage Control Point category. If necessary, change the timestep size of the Distributed Flow and Mean Precipitation slots (default is monthly). Input data into the Drainage Area, Curve Number, and Mean Precipitation slots on all control points and the Distributed Flow slot on the gage control point.

7.1.20.3 Drainage Area

This method computes the Distributed Flow values for all non-gage control points in the subbasin given the known Distributed Flow values of the Gage Control Point. There are no slots specifically associated with this method on the subbasin. The relevant data is held in method dependent slots on the control points and is accessed by the computational subbasin. See the Local Inflow Spatial Disaggregation on Subbasin method on the Control Point for details of these slots HERE (Section 9.1.3.3).

The Drainage Area method disaggregates using the drainage area ratio between the Ungaged Control Point and the downstream Gaged Control Point. The downstream Gage Control Point calculates the gain in the basin, using the difference in Distributed Flow between the downstream gage control point and its upstream gage control point(s) as defined in the downstream gage. Excluded gages may be defined in the Ungaged Control Point, to force the calculation to ignore the existence of that gage when solving the gain at the downstream control point.

The downstream Gage Control Point calculates the gain in the basin, using the difference in Distributed Flow between the downstream gage control point and its upstream gage control point(s) as defined in the downstream gage. The effect of excluded gages is defined below.

The downstream Gage Control Point calculates the area of the basin, subtracting the area of all upstream gages from the area provided in the downstream gage.

Excluded gages may be defined in the Ungaged Control Point, to change the gain calculation for its downstream gage control point. For most basins, the Excluded gages list is not needed. Each Ungaged Control Point maintains its own list of excluded gages. When a gage is placed in the list, the downstream gage ignores that excluded gage when solving the gain at the downstream control point.

The algorithm is described in the steps below:

1. All of the Ungaged Control Points in the computational subbasin are visited. For each Ungaged Control Point to be calculated, the Downstream Gage is visited, and the incremental flow for that gage is calculated. Distributed Flow from upstream gages is subtracted from the Distributed Flow in the currently computed gage. Any upstream gages that are not tributary to the ungaged point may be left in the gage (not subtracted) using Excluded Gages list. This list may be different for each Ungaged Control Point, based on its location in the subbasin. This set of incremental flows is only used in the remaining calculations pertaining to the current Ungaged Control Point.

2. The drainage areas are checked. An error occurs if the total drainage area of the Upstream Gage Control Points are equal or larger than the drainage area of the Downstream Gage Control Point or the Ungaged Control Point.

3. Incremental drainage area at the Downstream Gage Control Point is calculated, subtracting the sum of the drainage area for all Upstream Gage Control Points tributary to the Ungaged Control Point.

4. Incremental drainage area at the Ungaged Control Point is calculated, subtracting the sum of the Drainage Area for all Upstream Gage Control Points.

5. The ratio of incremental drainage area of the Ungaged Control Point to the Gaged Control Point is used to scale the downstream gage incremental flows.

6. Any upstream gages that are tributary to the Ungaged Control Point are added to the result of step 5.

7. The results are stored as input in the Ungaged Control Point’s Distributed Flow slot.

7.1.21 Local Inflow Temporal Disaggregation

The Local Inflow Temporal Disaggregation category contains two methods: the default, no action method, None, and the Specified Factors method. These methods are described below.

The selected method is executed at the beginning of the run if the subbasin is enabled. The three disaggregation methods are always executed in the following order: (1) Local Inflow Spatial Disaggregation, (2) Local Inflow Temporal Disaggregation, and (3) Incremental Local Inflows which may or may not include forecasting.

7.1.21.1 None

This method is the default for the Local Inflow Temporal Disaggregation category and should be selected when temporal disaggregation of local inflows is not desired. There are no slots specifically associated with this method.

7.1.21.2 Specified Factors

This method computes the Temporally Disaggregated Flow on all control points in the subbasin by multiplying the Distributed Flow value (monthly default) on each control point by the Temporal Disagg Factors (same timestep as run) located on the computational subbasin. There is one slot specifically associated with this method on the subbasin, the Temporal Disagg Factors slot. The other relevant data is held in method dependent slots on the control points and is accessed by the computational subbasin. See the Local Inflow Temporal Disaggregation on Subbasin method on the Control Point for details of these other slots.

The Specified Factors method sets the Temporally Disaggregated Flow slot on each control point to the product of the Distributed Flow value (on the control point) and the Temporal Disagg Factors value (on the subbasin). These values are set as input to avoid being cleared in subsequent model runs. After the temporal disaggregation is completed, all control points will contain a value in the Temporally Disaggregated Flow slot for all timesteps in the Temporal Disagg Factors slot.

The default timestep size for the Distributed Flow slot is monthly, however, this can be changed by the user by configuring the timeseries and is independent from the model timestep size. The user is responsible to ensure that the timestep size and the associated Temporal Disagg Factors data is appropriate. The timestep size for the Temporally Disaggregated Flow and Temporal Disagg Factors slots is always the same as that of the run control.

☞ **TEMPORAL DISAGG FACTORS**

Type: SeriesSlot
Units: FRACTION
Description: Proportion of the Temporally Disaggregated Flow on an object (usually daily) to the Distributed Flow on the object (usually monthly). Generally, the Temporal Disagg Factors should average to 1.0 over the Disaggregated Flow's timestep. For example, in the typical case, the daily Temporal Disagg Factors over any month should average to 1.0.
Information: Typically generated from historical data. The value must be greater than or equal to zero.
I/O: Required input
Links: Usually not linked

Model Setup

The computational subbasin must be set up for each subbasin in which flows will be temporally disaggregated. On the computational subbasin, select the Specified Factors method from the Local Inflow Temporal Disaggregation method category. Input data in the Temporal Disagg Factors slot on the computational subbasin. Append the relevant control points to the subbasin. Each temporal disaggregation subbasin should include all control points that will use the same Temporal Disagg Factors data for the disaggregation.

For each control point in the computational subbasin, the Specified Factors method must be selected from the Local Inflow Temporal Disaggregation category. If no values exist in the Distributed Flow slot (i.e., if spatial disaggregation was not/ will not be performed) enter the necessary data.

7.1.22 Incremental Local Inflows

The Incremental Local Inflows category is used to calculate the incremental local inflow to control points and reservoirs within the computational subbasin given the cumulative local inflows. The three disaggregation methods are always executed in the following order: (1) Local Inflow Spatial Disaggregation, (2) Local Inflow Temporal Disaggregation, and (3) Incremental Local Inflows which may or may not include forecasting.

7.1.22.1 None

This method is the default for the category and should be selected when cumulative local inflow data is not used or when calculation of incremental local inflow is not desired. There are no slots specifically associated with this method.

7.1.22.2 Compute Full Run Incremental Local Inflows method

This method, executed at the beginning of run for all timesteps in the run, calculates the incremental local inflows to all control points and reservoirs within the computational subbasin using the cumulative local inflows. There is no forecasting in this method.

There are no slots specifically associated with this method. The Cumulative Local Inflow and Incremental Local Inflow slots on control points and the Cumulative Hydrologic Inflow and Incremental Hydrologic Inflow slots on reservoirs as well as the routing method(s) in the intervening Reach(es) will be accessed during the calculation.

SOLUTION ALGORITHM:

Following is a description of the solution algorithm including step-by-step descriptions. The Compute Full Run Incremental Local Inflows method on the computational subbasin will execute at the beginning of a run and will set the new Incremental Local inflow slot on control points and the new Incremental Hydrologic Inflow slot on reservoirs with an Input flag. Because these slots are not dispatch slots, no object dispatching will occur. In practice, users will perform the calculation of incrementals, then save the model and disable the subbasins related to the calculation of incrementals. Thus, in future model runs the Compute Full Run Incremental Local Inflows method will not be executed. This practice allows users to avoid the computational expense of calculating the incremental local inflows at the beginning of each run. Should users wish to re-execute the Compute Full Run Incremental Local Inflows method, they should re-enable the subbasin and start a model run. The second, and subsequent times, the model is run, the calculated values that were set with an input flag will be overwritten with the new values.

The method calculates the Incremental Local/Hydrologic Inflow as the difference between a downstream control point or reservoir’s Cumulative Local/Hydrologic Inflow and the upstream control point's cumulative local inflow that is routed downstream through the intervening Reaches on that timestep. Following is the solution procedure.

SOLUTION STEPS

The method execute using the following steps; the details of these steps are described in the paragraphs below.

1. Group all control points, reservoirs and confluences in the subbasin into { U, D } pairs, where U is a control point upstream of D, and D is U’s closest downstream control point, reservoir, or confluence. If the Ignore Reservoirs boundary method is

selected, only control points and confluences are considered when creating the {U, D} pairs. For each { U, D } pair, repeat steps 2 - 5.

2. Obtain the upstream object’s Cumulative Local Inflow array (i.e., all timesteps). In the forecasting case, forecast the Cumulative Local Inflow using the method selected in the Generate Forecast Local Inflow category. Store the forecasted cumulative flow in the Forecasted Cumulative Local Inflow.

3. Route the upstream object’s Cumulative Local Inflow (or Forecasted Cumulative Local Inflow for forecasting) array through the intervening Reach(es) using the routing coefficients on the Reach(es).

4. If the downstream object is a confluence, store the routed array in its unused Temp Inflow{2,1} array. If the downstream object is a control point or reservoir, calculate the downstream object’s Incremental Local/Hydrologic Inflow at timestep t by subtracting the routed cumulative local inflow arriving downstream at time t from the downstream object’s cumulative local inflow at time t. Repeat for all timesteps t.

Incrementaldownstream(t)= Cumulativedownstream(t) - Routed Cumulativeupstream(t)

5. If the downstream object is a control point or a reservoir, set the downstream object’s Incremental Local/Hydrologic Inflow slot with the Input flag.

6. Move to next { U, D } pair. If no more pairs, move to step 7.

7. Group all confluence, control points, and reservoirs in the subbasin into { U, D } pairs, where U is a confluence upstream of D, and D is U’s closest downstream control point, reservoir, or confluence. If the Ignore Reservoirs boundary method is selected, only control points and confluences are considered when creating the {U, D} pairs. For each { U, D } pair, in upstream-to-downstream (partial-) order, repeat steps 8 - 11.

8. Add the upstream confluence’s Temp Inflow 1 array to its Temp Inflow 2 array and store the value in the Cumulative Local Inflow array.

9. Route the upstream object’s Cumulative Local Inflow array through the intervening Reach(es) using the routing coefficients selected on the Reach(es).

10. If the downstream object is a confluence, store the routed array in its unused Temp Inflow{2,1} array. If the downstream object is a control point or reservoir, calculate the downstream object’s Incremental Local/Hydrologic Inflow at timestep t by subtracting the routed cumulative local inflow arriving downstream at time t from the downstream object’s cumulative local inflow at time t. Repeat for all timesteps t.

Incrementaldownstream(t)= Cumulativedownstream(t) - Routed Cumulativeupstream(t)

11. If the downstream object is a control point or a reservoir, set the downstream object’s Incremental Local/Hydrologic Inflow slot with the Input flag.

12. Move to next { U, D } pair and return to step 8.

Step 1: Grouping Objects into Upstream Control Point - Downstream Object Pairs

The first step of the Compute Full Run Incremental Local Inflows method is to create internal tables of all control points, reservoirs and confluences in the subbasin. These tables are used to create a list of all pairs of objects { U, D }, where U is a control point upstream of D, and D is U's closest downstream control point, reservoir or confluence. If the Ignore Reservoirs boundary method is selected, only control points and confluences are considered when creating the {U, D} pairs. The computation of incremental local inflow (i.e., steps 2-5) will be performed on each of these { U, D } pairs. These pairs can be processed without regard to order.

Note that where confluences exist, the two upstream branches of the confluence must be processed prior to the downstream section of the confluence. For this reason, the Compute Full Run Incremental Local Inflows method will process all pairs with a control point upstream first (steps 2-5) and process all pairs with a confluence upstream second (steps 8-11). This ordering ensures that the proper data will be available to process the downstream section of the confluence.

Note that for the first (i.e., top-most) control point in the subbasin the Cumulative Local Inflow is also the incremental local inflow, thus no incremental flow must be computed for the top-most control point.

Step 2: Obtain Upstream Cumulative Local Inflow

For each { U, D } pair (the upstream object is a control point), the Compute Full Run Incremental Local Inflows method will obtain the data for all timesteps in the upstream object's Cumulative Local Inflow slot.

Step 3: Computation of Routing

Step 3 of the Compute Full Run Incremental Local Inflows method routes the Cumulative Local Inflow values through the intervening reaches. For this, it first finds all intervening reaches and grabs their routing coefficients, computing a composite routing vector as it goes. This assumes that all reaches use a linear routing method. Because of time lags, the routing must be done for all timesteps before moving to the next { U, D } pair.

Step 4: Calculate Incremental Local Inflow for Downstream Object

When the downstream object is a control point or a reservoir: After obtaining the upstream object's routed cumulative local inflow, the Compute Full Run Incremental Local Inflows method will calculate the incremental local inflow at timestep t for the downstream object by subtracting the routed cumulative local inflow arriving downstream at time t from the downstream object's Cumulative Local Inflow at time t (Eqn. 1). This is repeated for all timesteps in the Cumulative Local Inflow slot of the downstream object and the data is stored in the internal Incremental Local Inflow array. When the downstream object of the pair is a confluence: the Compute Full Run Incremental Local Inflows method does nothing at this step

Step 5: Set the Incremental Local/Hydrologic Inflow slot

The last step of processing the {Control Point, D} pairs is to set the appropriate slot(s) depending on the current pair's downstream object type (i.e., control point, reservoir or confluence). When the downstream object is a control point or a reservoir: the method will set the Incremental Local/Hydrologic Inflow slot with the Input flag to the values in the Incremental Local Inflow array. If the downstream object is reservoir, the method will set the Incremental Hydrologic Inflow slot with the Input flag to the values in the Incremental Local Inflow array. When the downstream object of the pair is a confluence: the Compute Full Run Incremental Local Inflows method stores the routed cumulative local inflow in the first unused Temp Inflow 1 or 2 slot on the confluence.

Step 6: Repeat for next pair

Step 7: Grouping Objects into Upstream Confluence- Downstream Object Pairs

After processing {Control Point, D} pairs, the Compute Full Run Incremental Local Inflows method uses its internal tables of all control points, reservoirs and confluences in the subbasin to create a list of all pairs of objects { U, D }, where U is a confluence upstream of D, and D is U's closest downstream control point, reservoir or confluence. The computation of incremental local inflow (i.e., steps 7-10) will be performed on each of these { U, D } pair. These pairs are processed in upstream-to-downstream (partial-) order. By the time any {Confluence, D} pair is processed, the confluence's two branches will have Temp Inflow 1 or 2 arrays filled from steps 2-5 or from execution of the following steps on an upstream pair.

Step 8: Obtain Upstream Cumulative Local Inflows

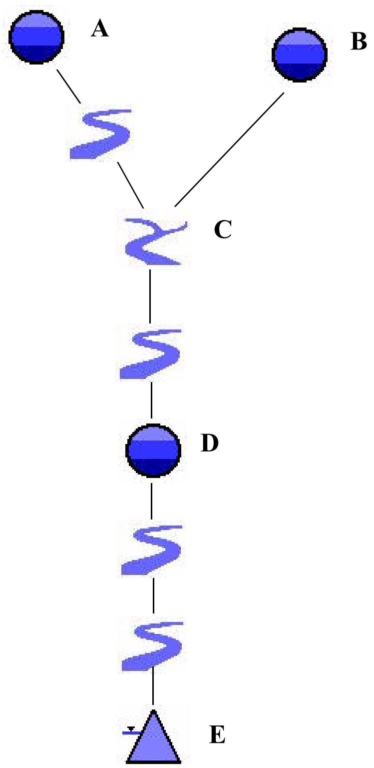
For each { U, D } pair (the upstream object is a confluence), the Compute Full Run Incremental Local Inflows method will sum the Temp Inflow 1 or 2 slots to obtain the values for the Cumulative Local Inflow array. If either or both of the Temp Inflow 1 or 2 slots does not contain values (i.e., one or both of the confluences upstream tributaries does not contain a control point), the values in the slot will default to 0.0.

Steps 9,10,11,12:

These steps are identical to Steps 3,4,5,6, above.

Example Schematic of method computation

A schematic of the computation is provided below:



* Route A's cumulative local inflow from A to C_{left} for all timesteps to obtain C_{left} routed cumulative local inflow.
* B's cumulative local inflow is C_{right} (no routing necessary).
* Sum C_{left} and C_{right} for all timesteps to get C_{sum}.
* Route C_{sum} to D for all timesteps.
* Compute incremental local inflow for D at each timestep as:

$$D_{(t)incremetal\ local} = D_{(t)cumulative\ local} - Csum_{(t)cumulative\ local\ routed}$$

* Route D's cumulative local inflow from D to E for all timesteps.
* Compute incremental local inflow for E at each timestep as:

$$E_{(t)incremetal\ local} = E_{(t)cumulative\ local} - D_{(t)cumulative\ local\ routed}$$

7.1.22.3 Compute Forecast Period Incremental Local Inflows

In the forecasting case, the Compute Forecast Period Incremental Local Inflows method on the Computational Subbasin will be executed at the beginning of each timestep to set the Local Inflow on each Control Point and the Hydrologic Inflow Forecast on each reservoir. Unlike the non-forecasting case, the subbasin and these methods must remain enabled for all runs as the local inflows are calculated on each timestep and are not given an input flag. NOTE: It is important that subbasins with Compute Forecast Period Incremental Local Inflows method selected stay enabled. This is because the forecasting takes place once each timestep over the forecast period. This is different than the application of the Compute Full Run Incremental Local Inflows for which users are encouraged to execute the method only once and then disable the subbasin.

The method first forecasts the Cumulative Local/Hydrologic Inflow throughout the forecast period and then calculates the Local Inflow or Hydrologic Inflow Forecast as the difference between a downstream control point or reservoir's Cumulative Local/Hydrologic Inflow Forecast and the upstream control point's forecasted cumulative local inflow that is routed downstream through the intervening reaches on that timestep.

There are no slots specifically associated with this method. The Cumulative Local Inflow slots on control points and the Cumulative Hydrologic Inflow slots on reservoirs as well as the routing method(s) in the intervening reach(es) will be accessed during the calculation. Method execution occurs at the beginning of a timestep for all enabled subbasins with the method selected.

Note: This method does not support the case where there is a routing reach directly below a confluence. Instead, insert a control point below the confluence, above the reach.

The use of this method for USACE-SWD models is presented HERE (USACE_SWD.pdf, Section 2.8).

Solution Algorithm

Check Method Selections: First the selected method will be initialized and the subbasins will be checked for errors. The Compute Forecast Period Incremental Local Inflows method will first check that every control point and reservoir in the subbasin has a forecasting method selected. All control points and reservoirs in the subbasin must have a forecasting method selected to execute the Compute Forecast Period Incremental Local Inflows method. If some of the objects have a forecasting method selected and some do not an error message is posted. The error message will state that all objects in the subbasin must either do forecasting or none of them may. If all of the reservoirs and control points in the subbasin have a forecasting method selected then the Compute Forecast Period Incremental Local Inflows must be executed.

The algorithm is very similar to the algorithm for the existing Compute Full Run Incremental Local Inflows method except it will execute on each timestep and compute forecasted incremental inflows throughout the forecast period. Following is a description of the steps this method executes. Please click HERE (Solution Steps) on page 105 for more information as the steps taken there are similar.

Group Pairs

The Compute Forecast Period Incremental Local Inflows method computes the forecasted incremental flows and sets the Local Inflow slot by looping through all pairs of objects in subbasin. It first groups all reservoirs, control points and confluences into {upstream, downstream} pairs and then process all {Control Point, D} pairs, next all {Reservoir, D} pairs and finally all {Confluence, D} pairs. Note, if a reservoir is a headwater reservoir (or has no upstream objects on which the flow is cumulative), the reserovir will not get processed as it is never the downstream object in a pair. In this case, it is advisable to create a separate computational subbasin of all headwater reservoirs and select the “Reservoirs Only” method on the subbasin. Click HERE (Section 7.1.23.4) for more information on this method.

Compute over Forecast Period

Instead of computing incrementals for the entire range of values in the Cumulative Local Inflow slot, the computation will be done only for the current timestep and the remaining timesteps in the forecast period.

Calculate Forecasted Cumulative Local Inflow (Reservoirs and Control Points upstream)

For the {Control Point, D} pairs and {Reservoir, D} pairs, the Compute Forecast Period Incremental Local Inflows method will call to the forecasting methods to calculate and set the slot Forecasted Cumulative Local/Hydrologic Inflow. Before calling to the forecasting method on the reservoir or control point, the method will first set the boolean argument calledFromSubbasin to TRUE. This will tell the forecasting method to recesses values of the Cumulative Local Inflow and set the Forecasted Cumulative Local/Hydrologic Inflow slot.

Route Forecasted Cumulative Local Inflow (Reservoirs and Control Points upstream)

Next, the Forecasted Cumulative Local/Hydrologic Inflow from the upstream object will be routed downstream through the intervening reaches resulting in an array of routed Forecasted Cumulative Local/Hydrologic Inflows. This step is analogous to the step in the existing Compute Full Run Incremental Local Inflows method that routes the Cumulative Local Inflow downstream.

Compute Forecasted Incremental Local Inflow (Reservoirs and Control Points upstream)

Next, the routed forecasted cumulative local/hydrologic inflow from the upstream object is subtracted from the Forecasted Cumulative Local/Hydrologic Inflow of the downstream object. This value is then entered in the Local Inflow slot (Hydrologic Inflow Forecast slot on the reservoirs if the Geometric Recession or Exponential Recession forecasting method is selected, Hydrologic Inflow slot on the reservoirs if the Coefficient and Exponent method is selected). This step is analogous to the step in the existing Compute Full Run Incremental Local Inflows method that subtracts the routed Cumulative Local Inflow of the upstream object from the Cumulative Local Inflow of the downstream object to set the Incremental Local Inflow slot. Note that in this application the Local Inflow slot is NOT set with the input flag, whereas in the non-forecasting method, the Incremental Local Inflow is set with the input flag.

The avoid reproducing code and to work with existing functions set up for the existing Compute Incremental Local Inflows method, an internal utility function, getSlotsForComputeIncrs, is used on the reservoir and the control point. This function return three arguments: upstream slot name, downstream slot name, and target slot. This function checks method selection on the control point or reservoir and sets the three arguments.

For the control point, if a forecasting method is selected and the Compute Forecast Period Incremental Flows method is selected, the getSlotsForComputeIncrs arguments are: Forecasted Cumulative Local Inflow (up), Forecasted Cumulative Local Inflow (down), and

Local Inflow (target). If no forecasting method is selected and the Compute Incremental Flows method is selected, the getSlotsForComputeIncrs arguments are: Cumulative Local Inflow (up), Cumulative Local Inflow (down), and Incremental Local Inflow (target).

For the reservoir, if the Geometric Recession or Exponential Recession forecasting method is selected and the Compute Forecast Period Incremental Flows method is selected, the getSlotsForComputeIncrs arguments are: Forecasted Cumulative Local/Hydrologic Inflow (up), Forecasted Cumulative Hydrologic Inflow (down), and Hydrologic Inflow Forecast (target). If the Coefficient and Exponent forecasting method is selected and the Compute Incremental Flows method is selected, the getSlotsForComputeIncrs arguments are: Forecasted Cumulative Local/Hydrologic Inflow (up), Forecasted Cumulative Hydrologic Inflow (down), and Hydrologic Inflow (target). If no forecasting method is selected and the Compute Full Run Incremental Flows method is selected, the getSlotsForComputeIncrs arguments are: Cumulative Hydrologic Inflow (up), Cumulative Hydrologic Inflow (down), and Incremental Hydrologic Inflow (target).

Process the Confluences

The algorithm for {Confluence, D} pairs is similar to the existing Compute Incremental Local Inflows algorithm. Instead of storing the routed Cumulative Local Inflow in the Temp Routed 1 or 2 slots on the confluence, this method will store the routed forecasted cumulative local inflow in the Routed 1 or 2 slots on the confluence. These two slots will then be summed as in the existing algorithm to calculate and set the Routed Outflow.

7.1.23 Reservoir Boundary for Incrementals

The Reservoir Boundary for Incrementals category on the computational subbasin is instantiated when one of the Incremental Local Inflows category methods are selected. The selected Reservoir Boundary for Incrementals method establishes whether the computation of incremental local inflows should stop at reservoirs or continue through reservoirs.

7.1.23.1 Stop at Reservoirs

This method is the default for the Reservoir Boundary for Incrementals category and should be selected when cumulative local inflow data is cumulative only until a reservoir is reached in a river system. When this method is selected the computation of incremental local inflows will stop at a reservoir and begin again below the reservoir. Between a reservoir and a downstream control point there will be no subtraction of cumulative local inflows to determine the incremental flow at the downstream control point. It is assumed that the cumulative local inflow data is also the incremental local inflow data for the first control point downstream of a reservoir in a subbasin.

7.1.23.2 Continue Through Reservoirs

This method should be selected when cumulative local inflow data is cumulative throughout the entire river system including reservoirs. When this method is selected the computation of incremental local inflows will continue through reservoirs. Between a reservoir and a downstream control point the reservoir's routed cumulative local inflow will be subtracted from the control point's cumulative local inflow to determine the incremental local inflow at the downstream control point.

7.1.23.3 Ignore Reservoirs

This method should be selected when cumulative local inflow data is cumulative throughout the entire subbasin but reservoir are not included, i.e. the reservoirs do not have any cumulative or incremental hydrologic inflows. When this method is selected, reservoirs do not need to be included in the subbasin and should not have an incremental flow method selected. If the reservoir is in the subbasin and has an incremental flow method selected, an error will be issued that method selection is inconsistent. When the incremental flow calculation is processing pairs of objects, it will skip the reservoir and find the next downstream object (either a control point or confluence).

7.1.23.4 Reservoirs Only

This method, available only when the Compute Forecast Period Incremental Local Inflows method is selected, should be selected when the user wishes to input local inflow data into the Cumulative Hydrologic Inflow slot on the reservoirs but the reservoirs do not have any upstream objects for which there are cumulative local inflows. This usually happens on headwater reservoirs but can also occur when a reservoir is directly downstream of another reservoir and inflows do not accumulate between the two reservoirs. This method allows users to input cumulative inflows consistently throughout the model in the Cumulative Hydrologic Inflow slot. Typically, the user would set up one subbasin containing all of the headwater reservoirs and other reservoir for which this applies. When the subbasin executes the Compute Forecast Period Incremental Local Inflows method, the Cumulative Hydrologic Inflow is forecasted into the Forecasted Cumulative Hydrologic Inflow and then copied to the Hydrologic Inflow Forecast slot.

7.1.24 Control Point Variable Routing Coefficients

This category is used to compute a set of control point Routing Coefficients based on the previous flows in the system. The category is dependent on having the **Operating Level Balancing** selected in the **Flood Control** category.

7.1.24.1 None

This method is the default for the Control Point Variable Routing Coefficients category and should be selected when a calculation of routing coefficients is not desired. There are no slots specifically associated with this method.

7.1.24.2 Compute Aggregate Coefficients

The **Compute Aggregate Coefficients** method allows the user to recalculate the Control Point routing coefficients based on flow in the reaches between the Control Point and its associated Reservoirs. There are no slots specifically associated with this method.

This method executes at the beginning of the timestep to recalculate routing coefficients for member Control Points based on the existing flows throughout the subbasin. It does the following:

```
ForEach Control Point in the computational subbasin:
       ForEach Reservoir in the Upstream Reservoir list slot:
              ForEach Reach between the current CP and the upstream reservoir, get the
              previous timestep's Inflow.
                     On the Reach.Variable Lag Coefficients slot, look-up the previous
                     Inflow on the column map to select the appropriate column of
                     coefficients to use. Store the column number for use below.
              EndFor
              When the "Compute Aggregate Coefficient" control point method is selected, if
              the looked-up column for the Inflow on each Reach is the 0th column
                     On the CP, copy the values from the Routing Coefficients to the
                     Computed Routing Coefficients. The standard values will be used.
              Else, recalculate the coefficients as follows (This is used for each timestep
              when the "Compute Aggregate Coeffs every Timestep" control point method is
              selected):
                     Starting at the upper most reach, route a unit Inflow (i.e. 1.0 cms)
                     through the reaches. Each reach must use the set of coefficients
                     determined from the inflow. Any unknown inflows are assumed to be 0.0.
                     The set of outflows at the lowest reach represents the coefficients.
                     Set the Computed Routing Coefficients table slot for the
                     appropriate reservoir
       EndFor, Move on to next Reservoir in upstream list and repeat
EndFor, Move on to next Control Point in the subbasin and repeat
```

In the above algorithm, the Inflow represents the flow for the previous timestep including previous surcharge forecasts, previous Flood Control releases, previous hydropower and any local flows.

The required Control Point methods are described **HERE (Section 9.1.19.2)** and the required reach methods are described **HERE (Section 22.1.1.6)** or **HERE (Section 22.1.21.2)**. For USACE-SWD models, this method is also discussed **HERE (USACE_SWD.pdf, Section 2.10)**.

7.1.25 Initialize Flow Slots for Routing

This category contains methods that are used to initialize slots that are required to be known for routing to solve at the start timestep. It sets values in these slots at pre-simulation timesteps according to the selected method. There are three methods: the default, no-action **None**, **Backcast Zeros** and **Backcast Initial Value**. No slots are added by any of these methods.

The methods are executed at the beginning of run. When either the **Backcast Zeros** and **Backcast Initial Value** are executed, an internal helper method is called that does the work. There are four parts to this helper method: 1) Loop through the Computational Subbasin's objects to identify slots of interest, 2) Resize the slots, if necessary, to include the earliest dispatch timestep, 3) Determine the value to set, and 4) Set values on slots, as required. These four steps are described below:

1. Loop through the Computational Subbasin's objects to identify slots of interest

The helper method considers each simulation object belonging to the Computational Subbasin and creates a list of slots that are required for initialization

The following slots are added if they are **not** linked to an upstream object, i.e. they are the headwater in the basin. Note, the slots consider their linking status to determine if they are headwater object. However, linking the slot to a DataObject, though not truly an upstream object, will cause the slot to be left off the list.

- Aggregate Reach.Inflow
- Confluence.Inflow1
- Confluence.Inflow2
- Control Point.Inflow
- Gage.Inflow
- Inline Power Plant.Inflow
- Reach Inflow

The following slots are added if the slot is linked. In the code, the slot at the downstream end of the link is added to the list of slots. Since the two slots are linked, values will propagate.

- Inline Pump.Outflow
- Pipeline.Outflow
- Reservoir.Outflow
- Reservoir.Seepage (when instantiated)

The following slots are added based on the specified conditions

- Reach.Diversion: added to the list if it is in use (i.e. instantiated by method selection) and linked.
- Reach.Return Flow: added to the list if it is in use and linked.
- Reach.Local Inflow: added to the list if it is in use.

Diagnostics for this specific process exist under Dispatch Management, SimObj.

2. Resize the slots

For each of the slots of interest, the helper method resizes the slots, as necessary, to include the first possible dispatch timestep. The first possible dispatch timestep is unique for each object and is determined as follows:

- Look for the earliest input on the current object and any linked upstream objects. Click **HERE (Simulation.pdfSimulation.pdf, Section 5.1)** for more information on this algorithm.
- For objects that are a part of a subbasin with either the **Backcast Zeros** or **Backcast Initial Value** method selected, the method will look downstream at each Reach object and sum the required number of routing timesteps. For each downstream Reach, the required number of pre-simulation timesteps is as follows:
    - No Routing: 0 timesteps
    - Step Response, Variable Step Response, Impulse Response routing:

      (Number of Lag Coefficients – 1)

    - Time Lag or Variable Time Lag: lag rounded up to the next timestep. (e.g. 36hr lag in daily model means 48 hrs = 2 days or 2 pre-simulation timesteps are required)

If a Reach does not have one of these routing methods selected, an error will be issued and the run will stop. The search is stopped when a reservoir or other object that does not dispatch before the start of the run is met or the downstream most point in the basin is found. The earliest timestep found by this algorithm is the first dispatch timestep for that object.

Note, this search algorithm is only valid for certain objects. The following objects will return an error if included in the subbasin as they have multiple outflows:

- Bifurcation
- Pipe Junction

Objects not typically on the main channel will not work in the search algorithm.

3. Determine the value to set

The helper method determines the value to use for initialization based on the selected method as described for either the **Backcast Zeros** or **Backcast Initial Value** method below.

4. Set values on slots

Based on the slot of interest and given the earliest dispatch timestep and an initialization value, a fairly simple algorithm is used to set those values on the slot. For each timestep between the earliest dispatch timestep and the initial timestep (inclusive), if the value on the slot is not valid, it is set to the initialization value. As always, these method will not overwrite any inputs that exist on the slot.

Note: When initializing Reservoir.Seepage, if the **Single Seepage Value** method is selected on the reservoir, at the beginning of the run, the **Seepage** is set equal to the **Single Seepage Value** if the initial **Seepage** is not input. Then when the **Backcast Initial Value** method executes, it will have an initial value that it will then backcast, no further inputs are required. If the user wishes to **Backcast Zeros**, they will need to input a 0 on the initial **Seepage**, otherwise the scalar value will be used for the initial timestep and zeros for all other pre-simulation timesteps. As always, inputs will not be overwritten by these methods.

7.1.25.1 None

This is the default, no-action method. This should be selected when the user wishes to explicitly input required initialization values directly onto the flow slots.

7.1.25.2 Backcast Zeros

The flow value 0.0 is used to initialize all required pre-simulation values. This should be selected when the user does not wish to input a value for the initial timestep and / or a flow of 0.0 is acceptable for initialization.

7.1.25.3 Backcast Initial Value

The earliest known pre-simulation value from the slot is used for initialization. The algorithm begins with the initial timestep and works backwards towards the earliest dispatch timestep. It finds the earliest valid value before encountering an invalid (NaN) value. The found value is returned for initialization. If no invalid value is found, the value at the earliest dispatch timestep is returned (although it will have no effect). If the value at the initial timestep is invalid, an abortive error is issued.

7.1.26 Groundwater Computation

This category is used to specify the setup for linked groundwater modeling in RiverWare, and the role that the computational subbasin plays in this modeling. Currently, the only groundwater model that can be linked with RiverWare is MODFLOW.

7.1.26.1 None

This is the default method. There are no slots associated with this method.

7.1.26.2 Link to MODFLOW 2000 RIP ET

Note: RiverWare's connection with MODFLOW is currently not functional. This method has been disabled and cannot be selected. An error will be posted at model load if this method was previously selected. Contact CADSWES for help.

When this method is selected eight additional categories are made available. These categories contain features that allow the user to dynamically link a RiverWare Model with a MODFLOW 2000 model. Click HERE (Section 7.2.1) to go to the RiverWare - MODFLOW Connection Functionality Guide. In order for the dynamic link between the programs to work, the user must select a method in certain categories. Following is a list of the available categories and whether the category requires a method selection.

- "Reach Stage" category - Required method selection
- "Reach Gain Loss" category - Required method selection
- "Groundwater Elevation" category - Required method selection
- "Groundwater Lateral Flux" category - Required method selection
- "WaterUser Surface Return Flow" category - Optional method selection
- "AggDiversion Site Surface Return Flow" category - Optional method selection
- "Reach Diversion" category - Optional method selection
- "Reach Local Inflow" categories - Optional method selection

Note: In MODFLOW cells and stream segments are indexed by layer, row, column and/or by segment number. In these eight categories, the user is able to select a method and can designate specific MODFLOW cells/segments as belonging to a simulation object which must be included in the computation subbasin. In cases where multiple cells correspond to the same object, the computational subbasin method will aggregate and disaggregate the cell values to the corresponding object as necessary.

SLOTS SPECIFIC TO THIS METHOD

☞ **NONE**

7.1.27 Reach Stage

This category is dependent on selection of the "Link to MODFLOW 2000 RIP ET" method in the "Groundwater Computation" category. This category should be used in conjunction with the “Reach Gain Loss” category on which a method other than “None” should be selected.

7.1.27.1 None

This is the default method. There are no slots associated with this method.

7.1.27.2 Weighted Interpolation

Note: RiverWare’s connection with MODFLOW is currently not functional. This method has been disabled and cannot be selected. An error will be posted at model load if this method was previously selected. Contact CADSWES for help.

This method calculates a stage elevation for each of the MODFLOW cells specified by the user in the Reach Stage and GainLoss Map slot. The stage is interpolated from between the Reach.Inflow Stage and Reach.Outflow Stage and is shown in the Reach Stage to MODFLOW slot. The calculated stage, for each cell in the Reach Stage to MODFLOW slot, is transferred to MODFLOW and used in the RIV package calculations. The interpolation equation is shown below. In some cases the river channel may span more than one cell in width, when this occurs the user may chose to set their weights so that all cells along a row will receive the same stage.

SLOTS SPECIFIC TO THIS METHOD

☞ REACH STAGE AND GAINLOSS MAP

Type: Table slot
Units: NO UNITS
Description: Maps a MODFLOW cell(s) to a Reach specified on the subbasin
Information: The layer, row, and column of each MODFLOW RIV cell is input by the user and the corresponding Reach object needs to be set as the row title. Interpolation weights corresponding to the Inflow Stage and the Outflow Stage on the Reach also need to be input for each MODFLOW RIV cell. The sum of the Inflow Stage Weigth and Outflow Stage Weight for a single cell should add up to 1.
I/O: Input only
Links: Not linkable

	Layer	Row	Column	Inflow Stage Weight	Outflow Stage Weight
Reach0	1	1	4	0.8	0.2
Reach0	1	2	4	0.5	0.5

	Layer	Row	Column	Inflow Stage Weight	Outflow Stage Weight
Reach1	1	4	6	0.7	0.3
Reach1	1	4	7	0.7	0.3

☞ Reach Stage to MODFLOW

Type: Table Series slot
Units: Length
Description: Interpolated stage (elevation) value for each MODFLOW cell specified in the Reach Stage and GainLoss Map slot. The column labels are automatically defined at the beginning of the run, in MODFLOW initialization. The user unit configuration (m, ft, etc.) selected on this slot must match with the units used in MODFLOW.
Information: Transferred to MODFLOW
I/O: Output only
Links: Not linkable

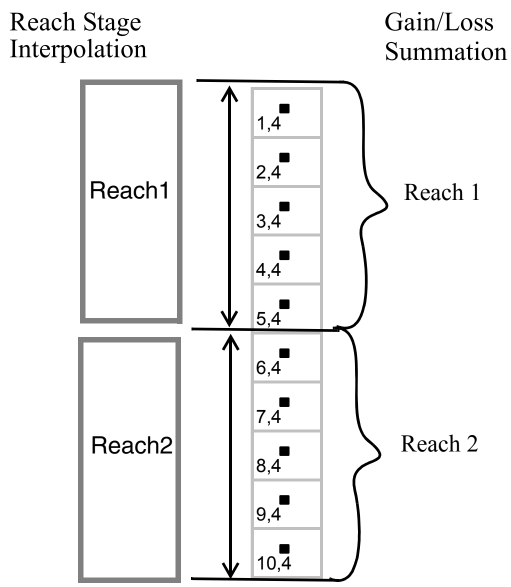
	Reach1 1,1,4 m	Reach1 1,2,4 m	Reach2 1,8,6 m	Reach2 1,9,7 m
t	553.2	552.7	551.4	550.9
t+1	553.32	552.5	551.7	550.8
t+2	553.27	552.6	551.5	550.9

Stage Interpolation Equation:

$$HRIV_n = w_1 StageU + w_2 StageD$$

$$\text{Where} \quad w_1 + w_2 = 1$$

$HRIV_n$ is the stage (water level elevation) in the river in MODFLOW aka the Reach Stage to MODFLOW (L)
$StageU$ is the Upstream Stage aka Reach.Inflow Stage (L)
$StageD$ is the Downstream Stage aka Reach.Outflow Stage (L)
w_1 is the inflow stage coefficient of interpolationn
w_2 is the outflow stage coefficient of interpolationn



Example Mapping of MODFLOW cells to Reach objects for Interpolation and Summation purposes

7.1.28 Reach Gain Loss

This category is dependent on selection of the “Link to MODFLOW 2000 RIP ET” method in the “Groundwater Computation” category. This category should be used in conjunction with the “Reach Stage” category on which a method other than “None” should be selected.

7.1.28.1 None

This is the default method. There are no slots associated with this method.

7.1.28.2 Summation

Note: RiverWare’s connection with MODFLOW is currently not functional. This method has been disabled and cannot be selected. An error will be posted at model load if this method was previously selected. Contact CADSWES for help.

The river GainLoss for each cell(s) in the Reach Stage and GainLoss Map slot is gathered from MODFLOW and set in the Reach GainLoss from MODFLOW slot. For each Reach, the individual gain/losses will be summed and mapped to the Reach.Total MODFLOW GainLoss slot. The summation equation is shown below.

SLOTS SPECIFIC TO THIS METHOD

☞ Reach Stage and GainLoss Map

Type: Table slot
Units: No Units
Description: Maps MODFLOW cell(s) to a Reach specified on the subbasin
Information: The layer, row, and column of each MODFLOW RIV cell is input by the user and the corresponding Reach object needs to be set as the row title. The sum of the Inflow Stage Weight and Outflow Stage Weight for a single cell should add up to 1.
I/O: Input only
Links: Not linkable

	Layer	Row	Column	Inflow Stage Weight	Outflow Stage Weight
Reach1	1	1	4	0.8	0.2
Reach1	1	2	4	0.5	0.5
Reach2	1	8	4	0.6	0.4
Reach2	1	9	4	0.3	0.7

☞ Reach GainLoss from MODFLOW

Type: Table Series slot
Units: Flow
Description: Each MODFLOW RIV cell specified in the Reach Stage and GainLoss Map slot contains a river GainLoss value transferred from MODFLOW. The column labels are automatically defined at the beginning of the run, MODFLOW initialization. The user unit configuration (cfs, ft3/day, etc.) selected on this slot must match with the units used in MODFLOW.
Information: Transferred to RiverWare from MODFLOW.
I/O: Output only
Links: Not linkable

GainLoss Summation Equation

$$\text{Reach.Total MODFLOW GainLoss} = \sum_{n=1}^{N} QRIV_n$$

n is the reach or cell number in MODFLOW
$QRIV_n$ is the flow between the river and the aquifer aka the Reach GainLoss from MODFLOW (L^3/T)

7.1.29 Groundwater Elevation

This category is dependent on selection of the “Link to MODFLOW 2000 RIP ET” method in the “Groundwater Computation” category. This category should be used in conjunction with the “Groundwater Lateral Flux” category on which a method other than “None” should be selected.

7.1.29.1 None

This is the default method. There are no slots associated with this method.

7.1.29.2 Weighted Interpolation

Note: RiverWare’s connection with MODFLOW is currently not functional. This method has been disabled and cannot be selected. An error will be posted at model load if this method was previously selected. Contact CADSWES for help.

This method calculates an head (elevation) for each of the MODFLOW cells specified by the user in the GroundWater Elevation Upstream Map and GroundWater Elevation Downstream Map slots using a weighted interpolation. An individual cell head is interpolated from between two GroundWater object’s heads (elevations) (GroundWater Storage.Elevation) and set in the GroundWater Elevation to MODFLOW slot. The calculated elevation for each cell in the GroundWater Elevation to MODFLOW slot is transferred to MODFLOW. The interpolation equation is shown below. The GroundWater Elevation Upstream Map and GroundWater Elevation Downstream Map slots should contain the same list of MODFLOW cells.

Slots Specific to this Method

☞ **GroundWater Elevation Upstream Map**

Type: Table slot
Units: No Units
Description: Maps a MODFLOW cell to a GroundWater Storage object specified on the subbasin
Information: The layer, row, and column of each MODFLOW GHB cell is input by the user and the corresponding upstream GroundWater Storage object needs to be set as the row title. Interpolation weights corresponding to the upstream GroundWater Storage object also need to be input. The sum of the Upstream Elevation Weight and Downstream Elevation Weight for a single cell should add up to 1.
I/O: Input only

Links: Not linkable

	Layer	Row	Column	Upstream Elevation Weight
GW 1	1	1	1	0.7
GW 1	1	2	1	0.5
GW 2	1	6	1	0.4
GW 2	1	7	1	0.2

☞ GroundWater Elevation Downstream Map

Type: Table slot
Units: No Units
Description: Maps a MODFLOW cell to a GroundWater Storage object specified on the subbasin
Information: The layer, row, and column of each MODFLOW GHB cell is input by the user and the corresponding downstream GroundWater Storage object needs to be set as the row title. Interpolation weights corresponding to the downstream GroundWater Storage object also need to be input. The sum of the Upstream Elevation Weight and Downstream Elevation Weight for a single cell should add up to 1.
I/O: Input only
Links: Not linkable

	Layer	Row	Column	Downstream Elevation Weight
GW 1	1	1	1	0.3
GW 1	1	2	1	0.5
GW 2	1	6	1	0.6
GW 2	1	7	1	0.8

☞ GroundWater Elevation to MODFLOW

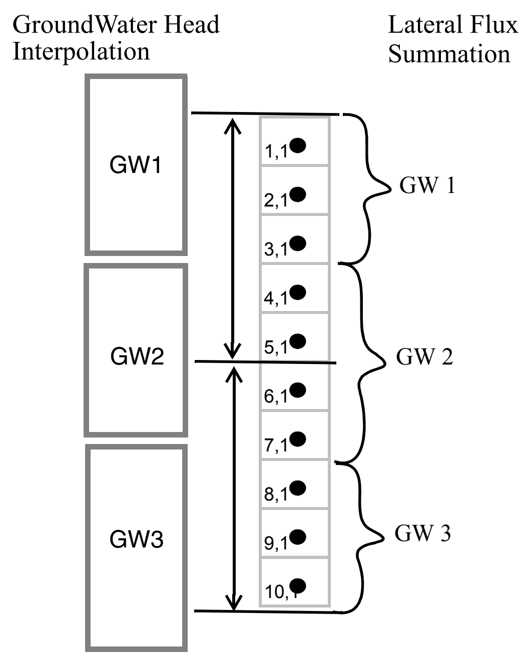
Type: Table Series slot
Units: Length
Description: Interpolated elevation (head) value for each MODFLOW cell as specified in the GroundWater Elevation Upstream Map and GroundWater Elevation Downstream Map slots. Value for each corresponding MODFLOW cell is interpolated between the Upstream and Downstream GroundWater objects. The column labels are automatically defined at the beginning of the run, in MODFLOW initialization. The user unit configuration (m, ft, etc.) selected on this slot must match with the units used in MODFLOW.
Information: Transferred to MODFLOW
I/O: Output only
Links: Not linkable

GROUNDWATER ELEVATION HEAD INTERPOLATION CALCULATION:

$$HB_n = w_1 Elevation_{GW}Up + w_2 Elevation_{GW}Down$$

$$\text{Where } \; w_1 + w_2 = 1$$

HBn	is the head (elevation, Bhead) assigned to the exteranal source in MODFLOW aka the GroundWater Elevation to MODFLOW
$Elevation_{GW}Down$	is the Groundwater Storage.Elevation on the Downstream GroundWater object (L) in RiverWare
$Elevation_{GW}Up$	is the Groundwater Storage.Elevation on the Upstream GroundWater object (L) in RiverWare
w_1	is the upstream elevation coefficient of interpolationn
w_2	is the downstream elevation coefficient of interpolationn



Example Mapping of MODFLOW cells to GroundWater Storage objects for Interpolation and Summation purposes

7.1.30 Groundwater Lateral Flux

This category is dependent on selection of the “Link to MODFLOW 2000 RIP ET” method in the “Groundwater Computation” category. This category should be used in conjunction with the “Groundwater Elevation” category on which a method other than “None” should be selected.

7.1.30.1 None

This is the default method. There are no slots associated with this method.

7.1.30.2 Summation

Note: RiverWare's connection with MODFLOW is currently not functional. This method has been disabled and cannot be selected. An error will be posted at model load if this method was previously selected. Contact CADSWES for help.

This method sums fluxes from the MODFLOW cells and writes the summed values into the RiverWare GroundWater Storage object. The GainLoss for each cell in the GroundWater Lateral Flux Map slot is gathered from MODFLOW and set in the GroundWater Lateral Flux from MODFLOW slot. For each GroundWater object the individual lateral fluxes will be summed and mapped to the GroundWater Storage.Lateral Flux from MODFLOW. The summation equation is shown below.

Slots Specific to this Method

☞ GroundWater Lateral Flux Map

Type: Table slot
Units: No Units
Description: Maps a MODFLOW cell to a GroundWater Storage object specified on the subbasin
Information: The layer, row, and column of each MODFLOW GHB cell is input by the user and the corresponding GroundWater object needs to be set as the row title.
I/O: Input only
Links: Not linkable

	Layer	Row	Column
GW 1	1	1	1
GW 1	1	2	1
GW 1	1	3	1
GW 2	1	4	1
GW 2	1	5	1
GW 2	1	6	1
GW 2	1	7	1

☞ GroundWater Lateral Flux from MODFLOW

Type: Table Series slot
Units: Flow
Description: Each MODFLOW GHB cell specified in the GroundWater Lateral Flux Map slot will contain a Lateral Flux value transferred to RiverWare from MODFLOW. The column labels are automatically defined at the beginning of

the run, in MODFLOW initialization. The user unit configuration (cfs, ft3/day, etc.) selected on this slot must match with the units used in MODFLOW.

Information: Transferred to RiverWare from MODFLOW
I/O: Output only
Links: Not linkable

LATERAL FLUX SUMMATION EQUATION

$$\text{Groundwater Storage.Lateral Flux from MODFLOW} = \sum_{n=1}^{N} QB_n$$

n is the reach or cell number in MODFLOW

QB_n is the flux into/out of cell i,j,k aka the GroundWater Lateral Flux from MODFLOW (L^3/T)

7.1.31 WaterUser Surface Return Flow

This category is dependent on selection of the “Link to MODFLOW 2000 RIP ET” method in the “Groundwater Computation” category.

7.1.31.1 None

This is the default method. There are no slots associated with this method.

7.1.31.2 One to One Exchange

Note: RiverWare’s connection with MODFLOW is currently not functional. This method has been disabled and cannot be selected. An error will be posted at model load if this method was previously selected. Contact CADSWES for help.

This method maps the Water User. Surface Return Flow to MODFLOW slot value to the WaterUser Surface Return Flow to MODFLOW slot for each corresponding segment designated in the WaterUser Surface Return Flow Map slot. The surface return flow values in the WaterUser Surface Return Flow to MODFLOW slot are transferred to MODFLOW and assigned as inflow into the specified segment.

SLOTS SPECIFIC TO THIS METHOD

☞ **WATERUSER SURFACE RETURN FLOW MAP**

Type: Table slot
Units: NO UNITS
Description: Maps a MODFLOW STR or SFR segment to a Water User object specified on the subbasin

Information: The segment number of each MODFLOW STR or SFR segment receiving a surface return flow is input by the user and the corresponding Water User object needs to be set as the row title.
I/O: Input only
Links: Not linkable

	Segment
WaterUser1	2
WaterUser2	13

☞ **WaterUser Surface Return Flow to MODFLOW**
Type: Table Series slot
Units: Flow
Description: Surface water return flow to each MODFLOW segment specified in the WaterUser Surface Return Flow Map slot. The column labels are automatically defined at the beginning of the run, MODFLOW initialization. The user unit configuration (cfs, ft3/day, etc.) selected on this slot must match with the units used in MODFLOW.
Information: Transferred to MODFLOW
I/O: Output only
Links: Not linkable

7.1.32 AggDiversion Site Surface Return Flow

This category is dependent on selection of the “Link to MODFLOW 2000 RIP ET” method in the “Groundwater Computation” category.

7.1.32.1 None

This is the default method. There are no slots associated with this method.

7.1.32.2 One to One Exchange

Note: RiverWare’s connection with MODFLOW is currently not functional. This method has been disabled and cannot be selected. An error will be posted at model load if this method was previously selected. Contact CADSWES for help.

This method maps the AggDiversion Site.Total Surface Return Flow slot value to the Agg Diversion Site Surface Return Flow to MODFLOW slot for each corresponding segment designated in the Agg Diversion Site Surface Return Flow Map slot. The surface return flow values in the Agg Diversion Site Surface Return Flow to MODFLOW slot will be transferred to MODFLOW and assigned as inflow into the specified segment.

Slots Specific to this Method

☞ Agg Diversion Site Surface Return Flow Map

Type: Table slot
Units: No Units
Description: Maps a MODFLOW STR or SFR segment to a AggDiversion Site object specified on the subbasin
Information: The segment number of each MODFLOW STR or SFR segment receiving a surface return flow is input by the user and the corresponding AggDiversion Site object needs to be set as the row title.
I/O: Input only
Links: Not linkable

	Segment
AggDiversion0	11
AggDiversion1	15

☞ Agg Diversion Site Surface Return Flow to MODFLOW

Type: Table Series slot
Units: Flow
Description: Surface water return flow to each MODFLOW segment specified in the Agg Diversion Site Surface Return Flow Map slot. The column labels are automatically defined at the beginning of the run, in MODFLOW initialization.The user unit configuration (cfs, ft3/day, etc.) selected on this slot must match with the units used in MODFLOW.
Information: Transferred to MODFLOW
I/O: Output only
Links: Not linkable

7.1.33 Reach Diversion

This category is dependent on selection of the “Link to MODFLOW 2000 RIP ET” method in the “Groundwater Computation” category.

7.1.33.1 None

This is the default method. There are no slots associated with this method.

7.1.33.2 One to One Exchange

Note: RiverWare's connection with MODFLOW is currently not functional. This method has been disabled and cannot be selected. An error will be posted at model load if this method was previously selected. Contact CADSWES for help.

This method maps the Reach.Diverion slot value to the Reach Diversion to MODFLOW slot for each corresponding segment designated in the Reach Diversion Map slot. The diverted flow value(s) in the Reach Diversion to MODFLOW slot are transferred to MODFLOW and assigned as inflow into the specified segment(s).

Slots Specific to this Method

☞ Reach Diversion Map

Type: Table slot
Units: No Units
Description: Maps a MODFLOW STR or SFR segment to a Reach object specified on the subbasin
Information: The segment number of each MODFLOW STR or SFR segment receiving a diversion inflow is input by the user and the corresponding Reach object needs to be set as the row title.
I/O: Input only
Links: Not linkable

	Segment
Reach2	5
Reach4	7

☞ Reach Diversion to MODFLOW

Type: Table Series slot
Units: Flow
Description: Diversion value for each MODFLOW segment specified in the Reach Diversion Map slot. The column labels are automatically defined at the beginning of the run, MODFLOW initialization. The user unit configuration (cfs, ft3/day, etc.) selected on this slot must match with the units used in MODFLOW.
Information: Transferred to MODFLOW
I/O: Output only
Links: Not linkable

7.1.34 Reach Local Inflow

This category is dependent on selection of the “Link to MODFLOW 2000 RIP ET” method in the “Groundwater Computation” category.

7.1.34.1 None

This is the default method. There are no slots associated with this method.

7.1.34.2 One to One Exchange

Note: RiverWare’s connection with MODFLOW is currently not functional. This method has been disabled and cannot be selected. An error will be posted at model load if this method was previously selected. Contact CADSWES for help.

A drain return flow for each segment specified in the Reach Local Inflow Map slot is gathered from MODFLOW and set in the Reach Local Inflow from MODFLOW slot. The Reach Local Inflow from MODFLOW slot on the subbasin is mapped to the specified Reach.Local Inflow MODFLOW Return slot.

Slots Specific to this Method

☞ **REACH LOCAL INFLOW MAP**

Type: Table slot
Units: NO UNITS
Description: Maps a MODFLOW STR or SFR segment to a Reach object specified on the subbasin
Information: The segment number of each MODFLOW STR or SFR segment receiving a diversion inflow is input by the user and the corresponding Reach object needs to be set as the row title.
I/O: Input only
Links: Not linkable

	Segment
Reach1	17
Reach3	18

☞ **REACH LOCAL INFLOW FROM MODFLOW**

Type: Table Series slot
Units: FLOW
Description: Each MODFLOW segment specified in the Reach Local Inflow Map slot contains a local inflow (return flow) transferred from MODFLOW to RiverWare. The column labels are automatically defined at the beginning of the run, MODFLOW initialization. The user unit configuration (cfs, ft3/day, etc.) selected on this slot must match with the units used in MODFLOW.

Information: Transferred to RiverWare from MODFLOW
I/O: Output only
Links: Not linkable

7.2 Functionality Guide

Following is a description of functionality associated with the Computational Subbasin (and therefore member objects) that spans mulitple user methods. Presented is a description, how to define a model, configure objects, and an example case for each of the functionalities. The Computation Subbasin specific categories and methods are described above. Currently, the only functionality described is the RiverWare-MODFLOW connection.

7.2.1 RiverWare - MODFLOW Connection

Note: RiverWare's connection with MODFLOW is currently not functional. The methods have been disabled and cannot be selected. An error will be posted at model load if methods were previously selected. Contact CADSWES for help.

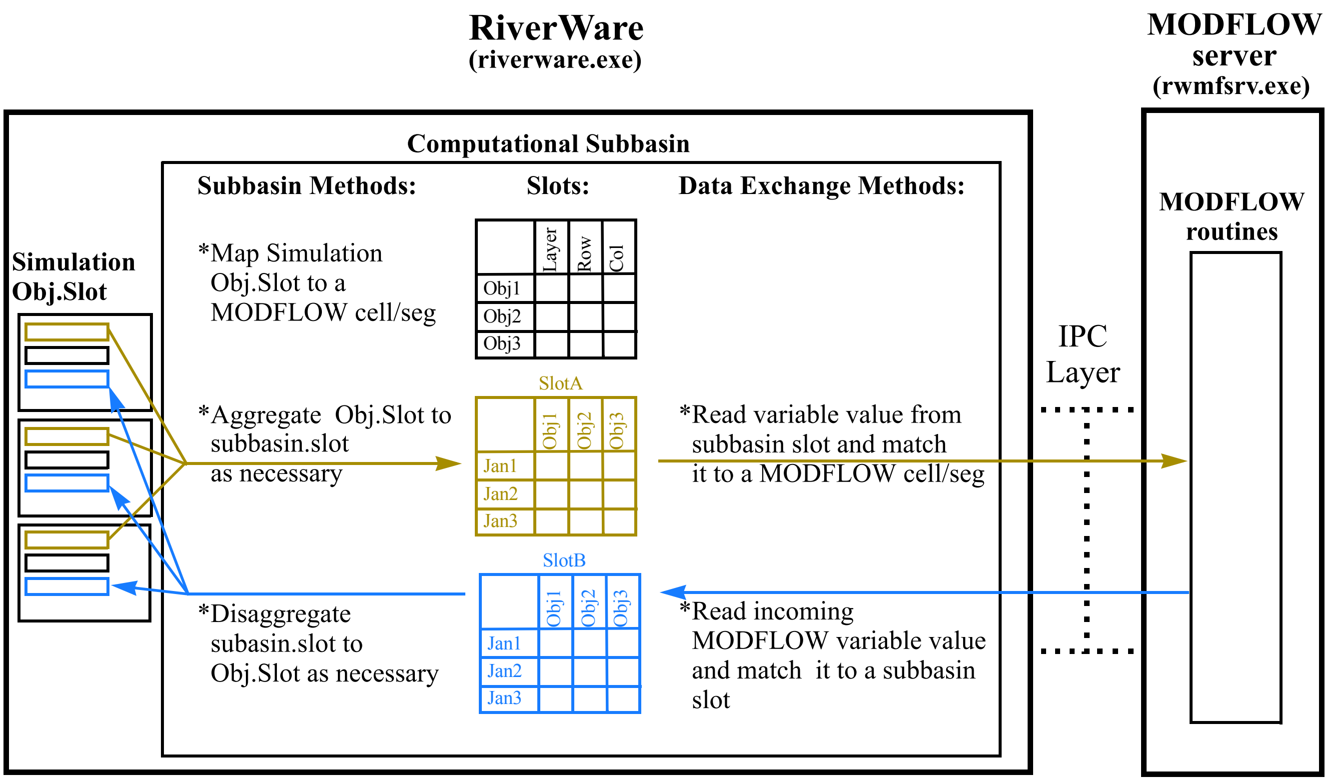
A dynamic link between RiverWare and MODFLOW 2000 (Harbaugh et al., 2000) allows the interaction between surface water and shallow groundwater to be incorporated into RiverWare. In RiverWare, the Reach, Water User, Aggregate Diversion Site (AggDiversion Site), and GroundWater Storage (GW) Objects each contain methods that, when selected, allow an object's data to be exchanged with MODFLOW. Data from these RiverWare objects is used as input into the MODFLOW General Head Boundary (GHB), River (RIV), Streamflow-Routing (STR), and updated Stream Flow Routing (SFR) packages. In the model setup, the user specifies the RiverWare objects and MODFLOW cell(s) or segment(s) involved in data exchange using the Computational Subbasin object. The Computational Subbasin displays cell by cell values for exchanged data. The MODFLOW 2000 executable provided with RiverWare includes one non-standard MODFLOW package Riparian Evapotranspiration (RIP-ET) (Maddock and Baird, 2003). While no data is exchanged between the RIP-ET package and RiverWare, this package is incorporated since it is applicable to regions where interaction between surface water and groundwater is highly active. This section describes how to set up a RiverWare-MODFLOW linked model.

7.2.1.1 How it Works

During a run in which RiverWare and MODFLOW are linked, the two models run in parallel, exchanging data at each timestep (the RiverWare timestep must match the MODFLOW stress period). To accomplish this interaction, when the user initiates a RiverWare-MODFLOW run RiverWare starts up a separate "MODFLOW server" process to perform the MODFLOW simulation. This executable is distributed and installed with RiverWare and contains all of the MODFLOW 2000 functionality, as well as the ability to communicate with RiverWare. This communication between RiverWare and the MODFLOW server is accomplished using an Interprocess Communication (IPC) mechanism appropriate for the platform on which RiverWare is running.

In RiverWare, the data exchange is defined and managed through the use of Computational Subbasins. The Computational Subbasin is a collection of objects that also contains methods and data used to define how information is exchanged between the two models. Figure 1 is a schematic of the overall data exchange managed by the Computational Subbasin.

Figure 1. Diagram of interaction between RiverWare objects, the computational subbasin, slots, servers, and MODFLOW routines



7.2.1.1.1 MODFLOW Library

The MODFLOW algorithms incorporated into the MODFLOW server executable was taken directly from the MODFLOW 2000 source code with some minor modifications. MODFLOW2000 introduced the concept of a process, the computations associated with a particular set of equations. The Parameter Estimation Process requires several MODFLOW “simulations,” and this functionality was achieved by introducing an outermost iteration into the MAIN procedure. A model which uses this process would not fit into the current framework -- RiverWare requires a MODFLOW model that iterates through the stress periods just as RiverWare iterates through timesteps. Thus in the modified MODFLOW subroutines, the parameter estimation loop was removed. Figure 5 in “Appendix A: MODFLOW Package Description” is taken from the MODFLOW 2000 documentation and illustrates the main flow of control for the MODFLOW program.

The MODFLOW libraries contained within the RiverWare server includes the following subroutines, several of which are primarily based on MODFLOW's MAIN procedure. They are listed in the order that they will be called by RiverWare:

1. Initialize RiverWare - Beginning of Run behavior

2. Initialize: This creates shared data structures of the appropriate size and reads data that applies to all timesteps.

- GLO DF, AL, and RP (DF = Define, AL = Allocate, RP = Read & Prepare)
- GWF AL
- OBS, SENS, and PES AL
- GWF RP
- OBS, SEN, and PES RP
- PES RW
- GWF AL and RP

3. "Begin Timestep" behavior in RiverWare

4. Begin MODFLOW Stress Period: Advance to the next stress period, read the appropriate data from RiverWare. Execute:

- GWF ST
- RP

5. Set RiverWare data: given data for the current stress period for a set of variables, use these values to override the values set by RiverWare during the "Begin Timestep" behavior.

6. Do MODFLOW Time Step Loop Period: perform the computation and output associated with the remainder of the current stress period. This is the timestep loop which is basically the remainder of the MAIN subroutine:

- GWF AD
- Iteration Loop
    - GWF FM
    - GWF AP
- GWF OC, BD, OT
- OBS FM
    - Parameter Sensitivity Loop
        - Iteration Loop
            - SEN FM
            - SEN AP
    - SEN OT

7. Get RiverWare data: get the MODFLOW values needed by RiverWare for the current timestep. Execute the RiverWare timestep, i.e. alternate between dispatching objects and firing rules as necessary.

8. Move on to the next timestep and repeat starting at step 3.

7.2.1.1.2 Data Exchange Slots

The Reach, GroundWater Storage, Water User and AggDiversion Site objects may all be linked with MODFLOW. The following table summarizes the data exchanged between the two models. Each RiverWare simulation object that contains exchanged data is listed by object type and slot name. The table indicates if the value shown on the Slot is a single value, a summed value, or an interpolated value (summed or interpolated values indicate that a RiverWare object may be associated with multiple MODFLOW cells). The direction of data exchange is noted and whether the data exchange is mandatory when linking the two models or if the data exchange is a user selectable option.

Table 2. Data exchanged between RiverWare and MODFLOW

Simulation Object	Slot	From	To	MODFLOW PACKAGE	MODFLOW Variable	Sum, Interpolation, or Single Value	MODFLOW Identifier
Reach	Total MODFLOW GainLoss	MODFLOW	RiverWare	RIV	River Leakage	SUM	Multiple cells (Layer,Row, Column)
Reach	Inflow Stage and Outflow Stage	RiverWare	MODFLOW	RIV	Stage	Interpolation	Multiple cells (Layer,Row, Column)
GroundWater Storage	Lateral Flux from MODFLOW	MODFLOW	RiverWare	GHB	Head Dep Bounds	SUM	Multiple cells (Layer,Row, Column)
GroundWater Storage	Previous Elevation	RiverWare	MODFLOW	GHB	Bhead	Interpolation	Multiple cells (Layer,Row, Column)
Reach	Local Inflow MODFLOW Return	MODFLOW	RiverWare	STR or SFR	Stream Flow Out or Stream Leakage (Flow Into Stream Reach)	Single Value	Segment #
Water User	Surface Return Flow	RiverWare	MODFLOW	STR or SFR	Flow	Single Value	Segment #
AggDivSite	Total Surface Return Flow	RiverWare	MODFLOW	STR or SFR	Flow	Single Value	Segment #
Reach	Diversion	RiverWare	MODFLOW	STR or SFR	Flow	Single Value	Segment #

At the beginning of each timestep [t], RiverWare transfers the following variables to MODFLOW; see Table 2 for a detailed description:

1. Reach Object: Inflow Stage and Outflow Stage at [t-1] (disaggregated for each MODFLOW cell)

2. Reach Object: Diversion at [t-1] (if Diversion goes to riverside drains or low flow channel)

3. Water User Object: Surface Water Return Flow [t-1]

4. AggDiversion Site Object: Total Surface Water Return Flow [t-1]

5. GroundWater Storage Object: Previous Elevation at [t] (same as Elevation at t-1) (disaggregated for each MODFLOW cell)

After sending this data to MODFLOW, RiverWare waits for MODFLOW to execute its timestep [t]. MODFLOW then sends data to the subbasin which writes values to the following RiverWare slots:

1. Reach Object: Total MODFLOW GainLoss at [t] (aggregated for each object)

2. Reach Object: Local Inflow MODFLOW Return at [t]

3. GroundWater Storage Object: Lateral Flux From MODFLOW [t] (aggregated for each object)

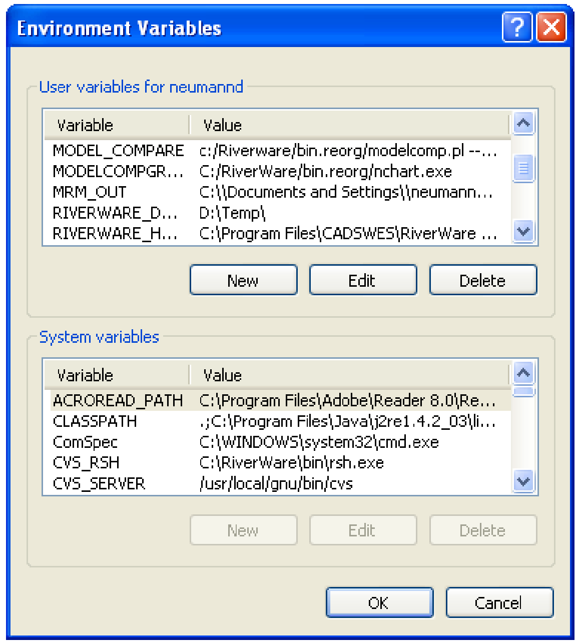
RiverWare then solves for timestep t using these values. Then the controller moves to [t+1].

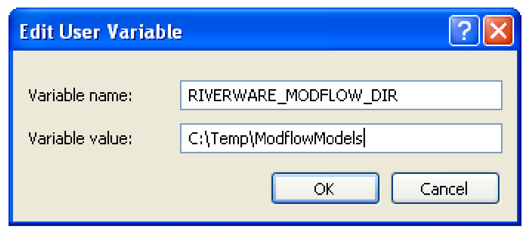
Note: The simulation timestep/stress-period size must be the same in RiverWare and MODFLOW.

7.2.1.2 Configuring RiverWare - Environment Variables and Diagnostics

First, a few environment variables need to be defined. Since MODFLOW is started from RiverWare, an environment variable must be set containing the PATH to the directory containing the MODFLOW.nam file(s).

- In Windows, from the Start menu, select **Control Panel ➥ System**, choose the **Advanced Tab**, then click on the **Environment Variable** button. Under the **User Variables** section select the **New** button and enter the following:
  - Variable Name: RIVERWARE_MODFLOW_DIR
  - Variable Path: (enter the path to the folder containing the MODFLOW.nam file(s), e.g. C:\Temp\ModflowModels)





In order to view the MODFLOW diagnostics in RiverWare a second environment variable will need to be set:

  - Variable Name: MODFLOW_SERVER_DIAG
  - Variable Path: 1 (or any other non-zero value)

- Diagnostics for the MODFLOW servers must also be enabled From the menu bar on the RiverWare Workspace select **Utilities ➥ Diagnostics Manager ➥ Workspace**, in the **Show Diagnostics For:** section put a check next to ‘**Client Server**’.

7.2.1.3 Model Setup

To begin, a MODFLOW model(s) should be created and tested. Some MODFLOW modeling details as they pertain to the MODFLOW RiverWare connection are provided here, but in general, it is assumed that the user already knows how to set up and use a MODFLOW model. In RiverWare, the user must decide how many and the type of objects they need. Since MODFLOW and RiverWare model spatial resolutions may differ, more than one MODFLOW cell can be associated with a RiverWare object. The user must decide what MODFLOW cells are associated with each object. A simple example Model is shown **HERE (Section 7.2.1.4)**.

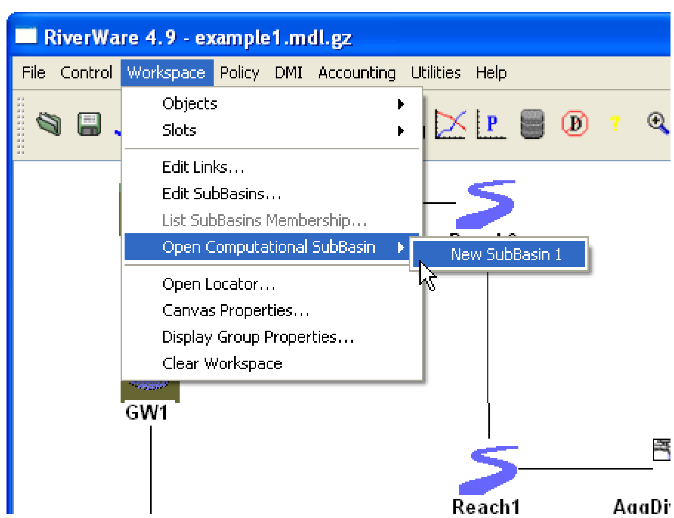
Input instructions for each object involved in the data exchange are discussed in detail in the following sections. These sections are broken into mandatory and optional data exchange configurations.

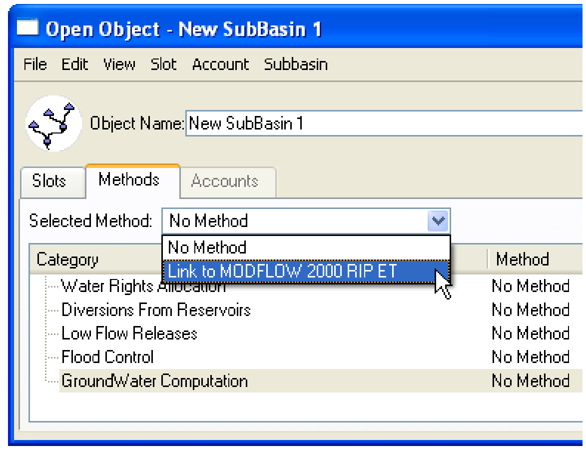
Note: In MODFLOW, the type and number of model cell boundary conditions should not change between stress periods. For example if 10 GHB cells are used in the first stress period the same 10 GHB cells must also be used in all stress periods in the model. This must be true of all boundary conditions in the model regardless if the boundary conditions are those that exchange data with RiverWare.

7.2.1.3.1 Computational Subbasin

A computational subbasin is a user configured subbasin that is used to specify a group of objects and the computations that should be performed on those objects. Once the user has set up their RiverWare simulation model, all simulation objects that are to exchange data with MODFLOW must be grouped into subbasins. Click **HERE (Subbasins.pdf, Section 1)** for more information on creating subbasins and adding objects to them.

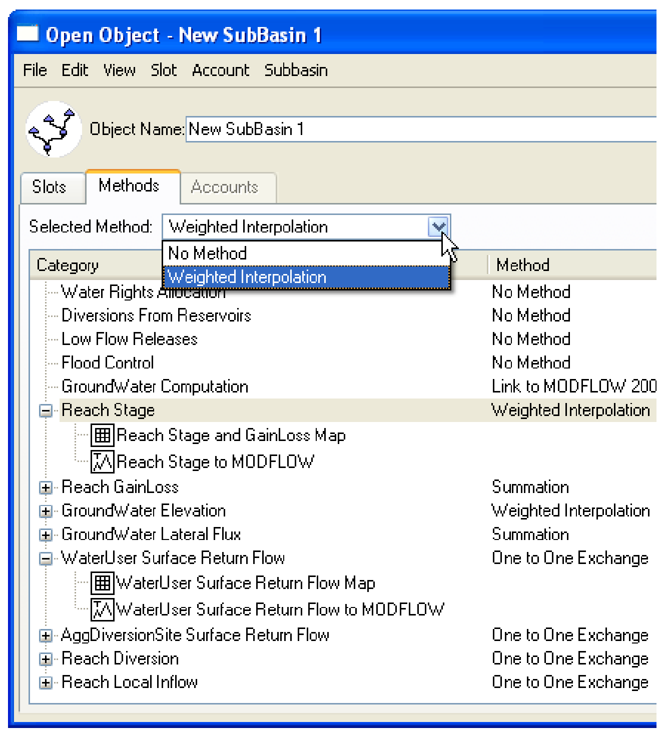
- Create a Computational Subbasin - The subbasin name must correspond to the MODFLOW model's ".nam" file.
- Add Objects to a Subbasin - All objects that exchange data with MODFLOW must be included in the computational subbasin. Each computational subbasin should connect with only one MODFLOW model.
- **Open the subbasin and configure it:** The computational subbasin object may be accessed in two ways: by clicking the **Open** button in the **Edit Subbasins** dialog, or by selecting **Workspace ➡ Open Computational Subbasin** from the main RiverWare menu.





The open subbasin dialog has two views: Slots and Methods. The slots view will be empty until a method has been selected. In the Methods view, highlight the “Groundwater Computation” category and select the “Link to MODFLOW 2000 RIP ET” method. Eight new categories are now available; select the desired methods from these categories:

- Reach Stage
- Reach Gain Loss
- Groundwater Elevation
- Groundwater Lateral Flux
- WaterUser Surface Return Flow
- AggDiversion Site Surface Return Flow
- Reach Diversion
- Reach Local Inflow

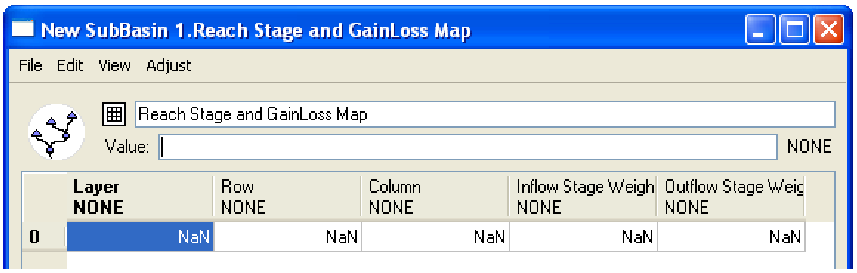


Each of the eight dependent categories have “None” as the default method. The “None” method has no associated slots and performs no computations. The methods in each of these categories preform different computations but the slot associated with each method are very similar. Each method contains a Map slot and a Data Exchange slot (except the Groundwater Elevation category which contains two Map slots and a Data Exchange slot). For example when the “Weighted Interpolation” method is selected in the “Reach Stage” category, the **Reach Stage and GainLoss Map** slot and the **Reach Stage to MODFLOW** slot are enabled. The user must supply input for the **Reach Stage and GainLoss Map**. This slot maps a Reach object’s data to one or more cells in a MODFLOW model.

The user must supply a list of the cells that exchange data with RiverWare. The **Reach Stage and GainLoss Map** slot contains five columns: Layer, Row, Column, Inflow Stage Weight, and Outflow Stage Weight. The Layer, Row, and Column correspond to the

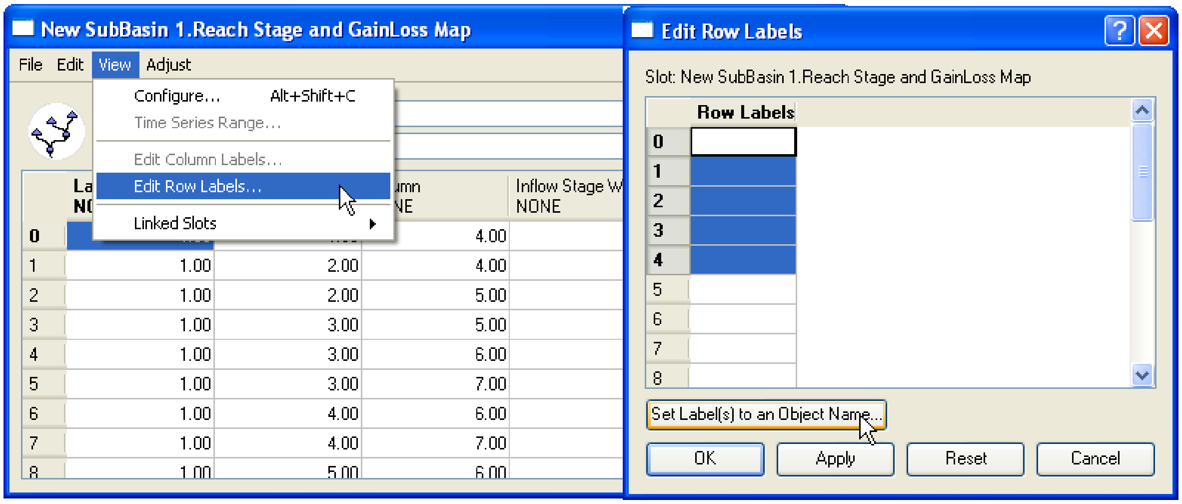
MODFLOW cell identification. The inflow and outflow stage weights are used in the interpolation equation (see the user help documentation included in the Appendix, **HERE (Section 7.2.1.6)**, for an explanation of the interpolation equation).

Note: The two weights corresponding to a single cell must add up to 1.



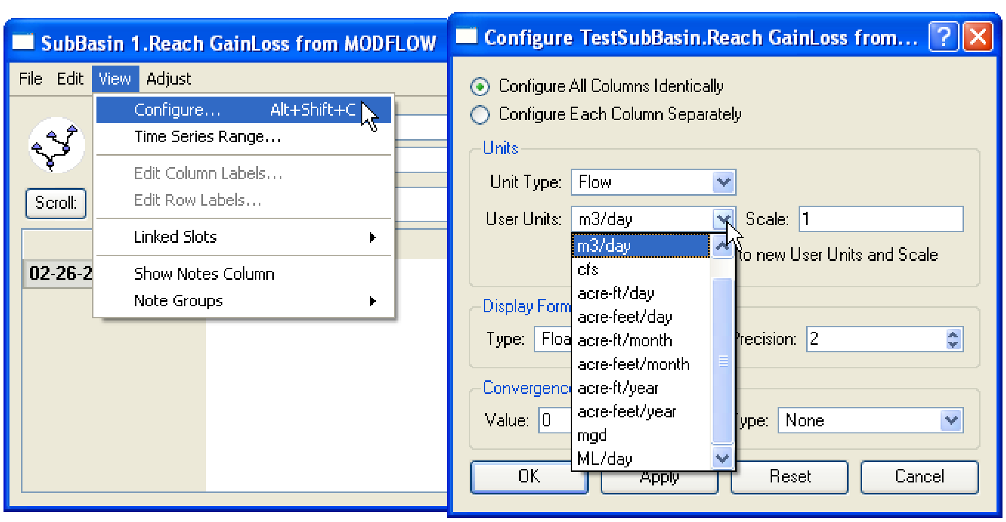
Likely it will be easiest for the user to import data into the Map slots. Imported data may be tab or space-separated. Imported data is assumed to be in the same units as shown in the Edit Slot dialog. The entire precision of an imported value will be preserved, although only the selected display precision is shown. Note: **Import (Fixed Size)** truncates incoming data if the data file contains more rows than the slot, and leaves existing data if the data file contains fewer rows than the slot. **Import (Resize)** automatically resizes the slot to match incoming data. The data will not be imported if the number of columns in the data does not match the number of columns needed in the slot or the data file contains unnecessary characters. More information on importing data can be found **HERE (Slots.pdf, Section 2.1.1)**.

Once the data is entered the user must edit the row labels. The rows should be labeled with the name of the corresponding object. From the slot dialog select **View ➥ Edit Row Labels**, highlight one or more rows in the **Edit Row Labels** dialog and push the **Set Label(s) to an Object Name** button, this invokes the object selector (initialized with the correct object type and only those objects in the subbasin), and highlight the corresponding object title and hit the Ok button in the object selector dialog. When finished assigning row titles, hit the Ok button in the Edit Row Labels Dialog.



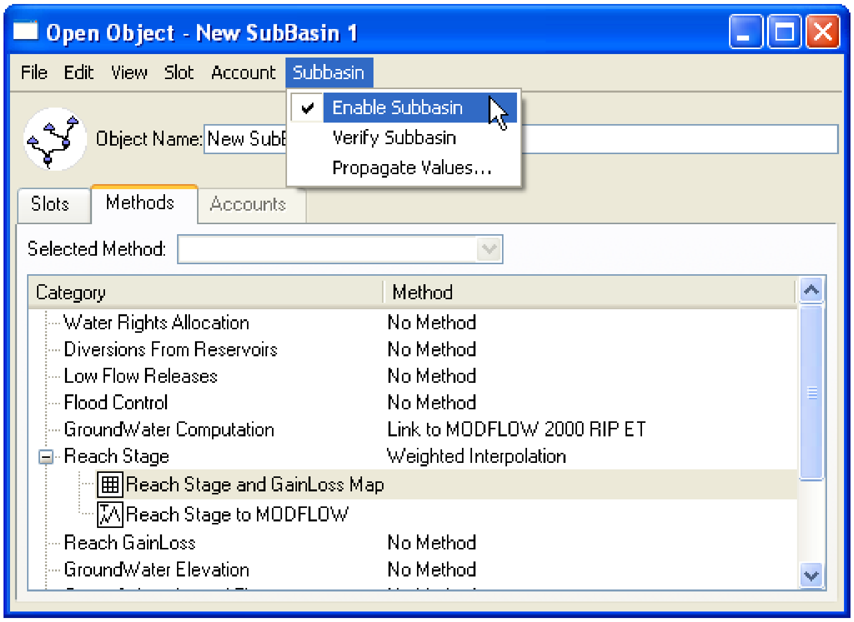
Data must be arranged so that all the cells pertaining to one object appear in contiguous rows. For example the user is not allowed to have rows 0-4 correspond to Reach 0, rows 4-8 correspond to Reach 1, and then row 9 again correspond to Reach 0. Instead the user should have rows 0-5 correspond to Reach 0 and rows 5-9 correspond to Reach 1.

Once the user has set up the Map slots, the Data Exchange slot(s) units must be configured. The units in the Data Exchange slots must be consistent with the units used in MODFLOW model(s). To set the corresponding units, select **View ➥ Configure** from the slot menu. Then, set the units for all columns at once:



Units must be set for the following Data Exchange slots the Reach Stage, Reach GainLoss, GroundWater Elevation, GroundWater Lateral Flux, WaterUser Surface Return Flow, Agg Diversion Site Surface Return Flow, Reach Diversion, and Reach Local Inflow when in use.

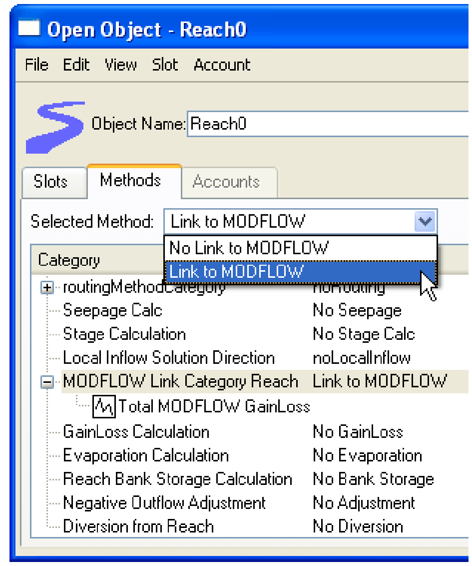
- **Enabling Subbasin**: Once the subbasin has been configured, make sure it is enabled. To enable a subbasin select **Subbasin ➥ Enable Subbasin** in the open subbasin object. Note: the subbasin will not perform any computations if it is not enabled.



7.2.1.3.2 Reach Object

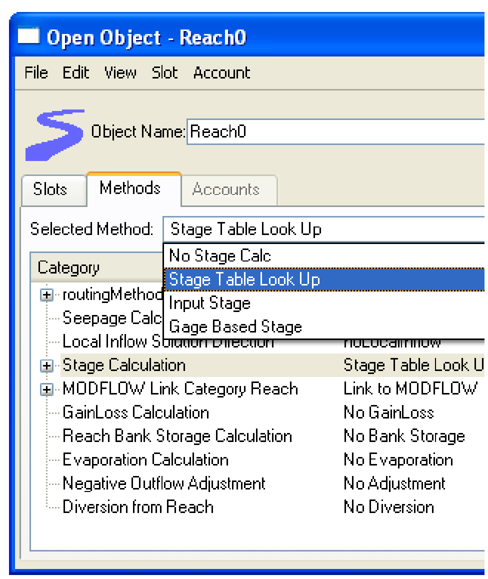
In RiverWare a river channel can be represented with a Reach object. When a MODFLOW model is connected with RiverWare the same river channel should be represented using a RIV type boundary condition in MODFLOW. As shown in Table 2, some of the data exchanged between the two models is mandatory and some optional. The first subsection below describes mandatory data, this data is necessary for the linked models to run and the user must select and assign values as described. The second subsection describes two optional data exchanges. None, one, or both of the optional exchanges may be assigned on a single Reach.

Mandatory Data Exchange: Reach Configuration



- **MODFLOW Link Category:** For each Reach that is to transfer data between RiverWare and MODFLOW, the user must select the “Link to MODFLOW” method in the “MODFLOW Link Category Reach”, **HERE (Section 22.1.20.2)**. The “MODFLOW Link Category Reach” is dependent upon selection of the No Routing, Time Lag, Variable Time Lag, or Muskingum Cunge method in the Routing category. When the “Link to MODFLOW” method is selected, the Total MODFLOW GainLoss slot becomes available. The MODFLOW RIV boundary condition calculates flow into or out of a cell from an external source in proportion to the difference between the head in the cell and the stage in the river. The value shown in the Total MODFLOW GainLoss slot is the sum of this flux over all the cells associated with a Reach. A negative flux on the Reach.Total MODFLOW GainLoss slot indicates flow out of the river to the aquifer. The user must assign MODFLOW cells to a Reach using the Computational Subbasin structure, **HERE (Section 7.1.28)**.

  The “MODFLOW Link Category Reach” also contains the method “No Link to MODFLOW”. This method should be selected when the user does not wish to link a Reach with MODFLOW. Note: A RiverWare model may contain both Reach objects that are linked to MODFLOW and Reach objects that are not linked to MODFLOW



- **Stage:** A Stage calculation method must be selected in the “Stage” category, **HERE (Section 22.1.6)**. The MODFLOW RIV boundary condition calculates flow into or out of a cell from an external source in proportion to the difference between the head in the cell and the stage in the river. To calculate this gain/loss between the river and the underlying aquifer, MODFLOW must have a stage value for each corresponding RIV boundary cell. The user must assign an initial inflow and outflow, as well as, initial inflow stage and outflow stage to every Reach that is linked with MODFLOW. All inflow and outflow stages proceeding the initial input will be calculated from the previous timesteps inflow and outflow by RiverWare for each RIV boundary cell and transferred to MODFLOW. The user must assign MODFLOW cells to a Reach from the Computational Subbasin.

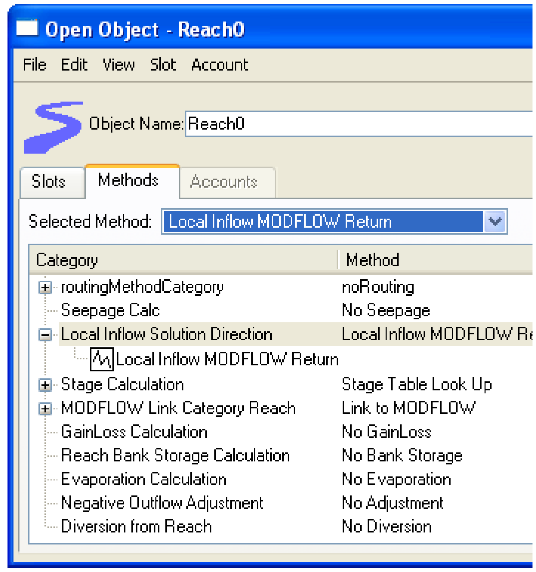
Note: In MODFLOW the user must have the option flag (CBC) entered in Line 2 of the RIV package input file. If this flag is not turned on, space is not allocated for the RIV package in the MODFLOW internal array and RiverWare cannot access the exchanged variables

Optional Data Exchange Reach Configuration

A surface water body (e.g. riverside drain) could be represented in MODFLOW using the STR or SFR packages. When a surface water body is defined in this manner it does not need to be explicitly represented in RiverWare. Flow between this type of MODFLOW surface water body, and a RiverWare Reach is optional. The user may choose to assign a return flow from a MODFLOW surface water body to a RiverWare Reach and/or a diversion from a RiverWare Reach to a MODFLOW surface water body.

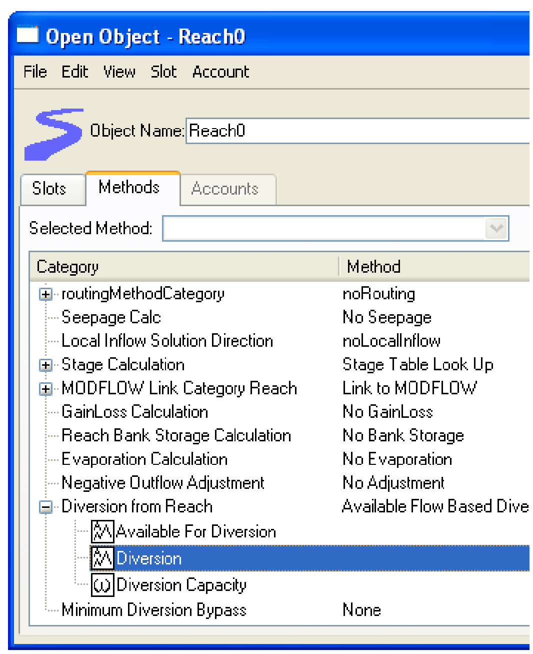
Note: In the MODFLOW STR and SFR package, stream/river/drain networks are assembled using reaches and segments. Reaches are joined together to form segments and all reachs in a segment share the same model properties. A reach can span up to one model cell, while segments can span multiple cells. Segments are numbered sequentially starting with the most upstream segment. The reaches in a segment are also numbered sequentially starting at the most upstream reach. Inflow and outflow along a MODFLOW stream network (STR or SFR) can only occur on the first and last reaches in a segment, respectively (with one exception in the SFR package, see Purdic et al., 2004). This must be considered when the user matches corresponding variables between a RiverWare Reach and a MODFLOW segment. It is suggested that the user create tributary and diversion segments that are to be used only for data exchange with RiverWare.

- **Local Inflow**: A return flow from a surface water body represented in MODFLOW (e.g. riverside drain) to a RiverWare Reach is optional. This return flow must be diverted from a MODFLOW STR or SFR segment and can be specified as a local inflow into a RiverWare Reach. The user needs to select the “Local Inflow MODFLOW Return” method in the “Local Inflow and Solution Direction” category, HERE (Section 22.1.3.8). The slot Local Inflow MODFLOW Return, is associated with this method and displays the value transferred from MODFLOW. The STR or SFR segment in MODFLOW providing the return flow is assigned using the computational subbasin structure. To avoid potential issues with data transfer, one RiverWare Reach is limited to connect with one MODFLOW segment.



Note: While a Reach that is assigned to receive a MODFLOW return flow may be linked with the MODFLOW RIV package (see Mandatory Data Exchange Reach Configuration) this is not required and the user may set the RiverWare Reach to receive only the Local Inflow MODFLOW Return. When this occurs the Reach.MF GainLoss slot will contain NaN’s.

- **Diversions:** A diversion from a Reach to a surface water body in MODFLOW (e.g. riverside drain) is optional. The method selection for a diversion is the same whether or not the user intends to divert to another RiverWare Object or MODFLOW. The user needs to select a method in the “Diversion from Reach”, HERE (Section 22.1.17), category and input a value into the Diversion slot (shown to the right). The diversion from the Reach is assigned as inflow into a MODFLOW STR or SFR segment. The user assigns the MODFLOW segment to receive the inflow using the computational subbasin structure. If no MODFLOW segment is assigned to the diversion, the user may link the Diversion as an inflow into another RiverWare object. Flow may only be diverted



from a Reach to one object at a time (e.g. one Reach cannot divert flow to both a MODFLOW segment and a RiverWare Water User).

Note: A Reach that has a Diversion to MODFLOW assigned may be linked with the MODFLOW RIV package (see Mandatory Data Exchange Reach Configuration) however this is not required and the user may set the Diversion as the only data exchange with MODFLOW. When this occurs the Reach.MF GainLoss slot will contain NaN’s.

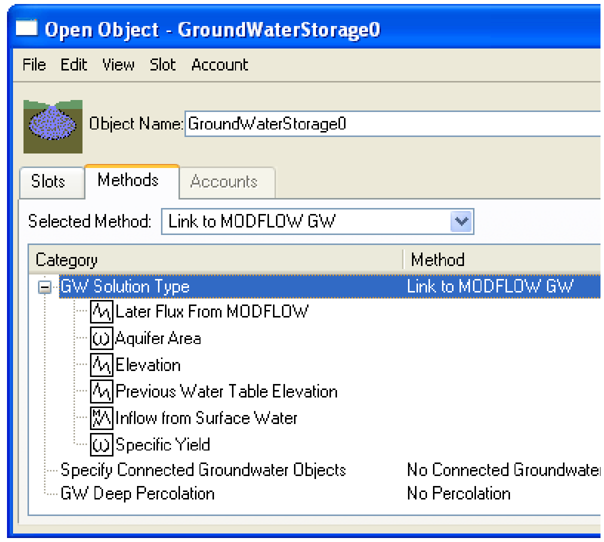
7.2.1.3.3 GroundWaterStorage Object

In a RiverWare-MODFLOW linked model, the GroundWater Storage (GW) object is used to incorporate boundary conditions into the RiverWare model and to manage GW return flows from RiverWare Water Users. Flux between the GW object’s head and head in the MODFLOW aquifer is computed for the lateral MODFLOW boundaries in MODFLOW. In addition to this flux, two other fluxes are calculated in RiverWare and are accounted for in the GW object’s storage equation. The second flux, is the flux between linked GW objects. The third flux, is a head-based flux between a GW object and a low-resolution MODFLOW model. The low resolution MODFLOW model head need to be input by the user into RiverWare using the Head Based Percolation method. As shown in Table 2 all of the data exchanged for the GW object is mandatory.

Note: In MODFLOW, boundary conditions are a necessary component in the MODFLOW model. Information from a low-resolution regional-scale groundwater model may be interpolated in space and/or time, by the user for inclusion directly in both the RiverWare and MODFLOW models. Along the lateral MODFLOW boundaries in the first layer the user should assign GHB boundary conditions where the elevation from RiverWare GW objects is used as the external source head in the GHB calculations. All other MODFLOW model edge boundaries may be assigned as the user chooses, since they do not require an interaction between RiverWare and MODFLOW and are not discussed in this document.

MANDATORY DATA EXCHANGE REACH CONFIGURATION

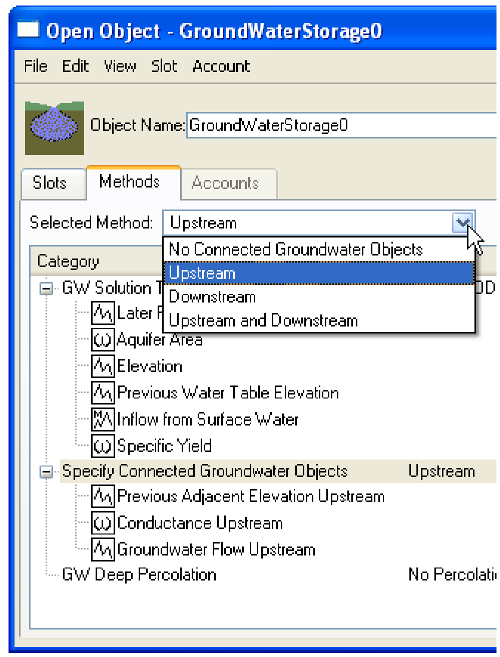
- **GW Solution Type**: The user must select the “Link to MODFLOW GW”, HERE (Section 14.1.1.4), method in the “Solution Type”, for the GW object to exchange data with MODFLOW. Six slots are associated with the Link to MODFLOW GW method:
  - Aquifer Area
  - Elevation
  - Previous Water Table Elevation
  - Inflow from Surface Water
  - Specific Yield
  - Lateral Flux from MODFLOW



The first five slots are also used by other "Solution Type" methods. The last slot Lateral Flux from MODFLOW is specific to the Link to MODFLOW GW method. Cells associated with the first boundary type or lateral boundary should be represented in MODFLOW using the GHB boundary condition. The GHB package simulates flow into or out of a cell from an external source in proportion to the difference between the head in the cell and the head assigned to the external source. To calculate this flux, MODFLOW needs an external head value for each GHB boundary cell. The user is required to assign an initial Elevation to every GW object that is linked with MODFLOW. The initial elevation head for each corresponding GHB boundary cell is calculated by interpolating the elevation between two GW objects in RiverWare. The user can assign upstream and downstream GW objects to individual cells using the computational subbasin HERE (Section 7.1.29). In addition to an initial Elevation, the user is required to assign an Aquifer Area and Specific Yield on each GW object. Optionally, the user may link a GW objects' Inflow from Surface Water slot with another RiverWare object.

Two other methods are available in the "Solution Type", these methods should be selected when the user does not wish to link a GW object with MODFLOW. Note: one model may contain both GW objects linked to MODFLOW and GW objects that are not linked to MODFLOW.

Note: In MODFLOW the user must have the option flag (CBC) entered in Line 2 of the GHB package input file. If this flag is not turned on space is not allocated for the GHB package in the MODFLOW internal array and RiverWare cannot access the exchanged variables.



- **Specify Connected Groundwater Objects**: The second flux, is the flux between linked GW objects. The user may choose to link a GW object with none, one or two other GW objects. The methods available in the “Lateral Link Direction” category, **HERE (Section 14.1.12)**, are: No Connected Groundwater Objects, Upstream, Downstream, and Upstream and Downstream (see below). When a method other than No Connected Groundwater Objects is selected, the following slots are available: Previous Adjacent Elevation, Conductance, and Groundwater Flow. The user is required to assign a Conductance between the two objects and must link the Previous Adjacent Elevation of one object to the Previous Elevation in the other object. For example if GW1 and GW2 are to be linked and GW1 is the upsteam of GW2. Then the Previous Elevation slot on GW1 needs to be linked with the Previous Adjacent Elevation Upsteam slot on GW2, and the Previous Elevation Downstream slot on GW1 needs to be linked with the Previous Elevation slot on GW2.

- **GW Deep Percolation**: The third flux, is a head-based flux between the GW object elevation and a low-resolution MODFLOW model head elevation. This flux can be represented using the “Head Based Percolation” in the “GW Deep Percolation” category, **HERE (Section 14.1.4.5)**. The method contains three slots: Percolation, Deep Aquifer Conductance, and Deep Aquifer Elevation. The Deep Aquifer Elevation represents the MODFLOW head elevation (from a low-resolution MODFLOW model) and must be input by the user. The user also needs to input a Deep Aquifer Conductance value for each GW object.

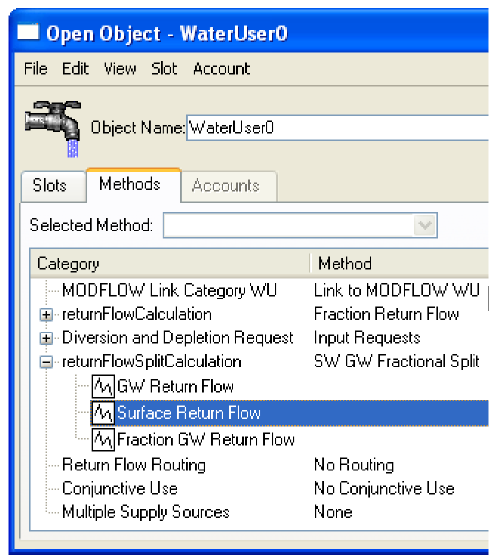
7.2.1.3.4 Water User

Linking a Water User object to MODFLOW is optional. The user may link none, some, or all the water user objects in a RiverWare model to MODFLOW.

Optional Data Exchange Water User Configuration

A surface water body (e.g. riverside drain) can be represented in MODFLOW using the STR or SFR packages. When a surface water body is defined in this manner it does not need to be explicitly represented in RiverWare. Flow between this type of MODFLOW surface water body and a RiverWare Water User is optional. The user may choose to assign a surface return flow from a Water User to a surface water body represented in MODFLOW (e.g. riverside drain).

Note: In the MODFLOW STR and SFR package, stream/river/drain networks are assembled using reaches and segments. Reaches are joined together to form segments and all reachs in a segment share the same model properties. A reach can span up to one model cell, while segments can span multiple cells. Segments are numbered sequentially starting with the most upstream segment. The reaches in a segment are also numbered sequentially starting at the most upstream reach. Inflow into a MODFLOW stream network (STR or SFR) can only occur on the first reach in a segment, respectively (with one exception in the SFR package). This must be considered when the user matches corresponding variables to the Water User. It is suggested that the user create tributary and diversion segments that are to be used only for data exchange with RiverWare.



- **Surface Return Flow**: A surface return flow from a water user to a surface water body in MODFLOW (e.g. riverside drain) is optional. The user needs to select the "Link to MODFLOW WU" method in the "MODFLOW Link Category WU" category, HERE (Section 27.1.11.2). The Surface Return Flow is calculated when methods are selected in both the Return Flow and Return Flow Split categories. The Surface Return Flow from the Water User is assigned as inflow into a MODFLOW STR or SFR segment. The user assigns the segment to receive the return flow using the computational subbasin structure. If no MODFLOW segment is assigned, the user may link the surface return flow as an inflow into another RiverWare object. The user may input a value for the Surface Return Flow at the initial timestep if desired, otherwise an initial inflow value of zero will be transferred to MODFLOW for use in the first timestep.

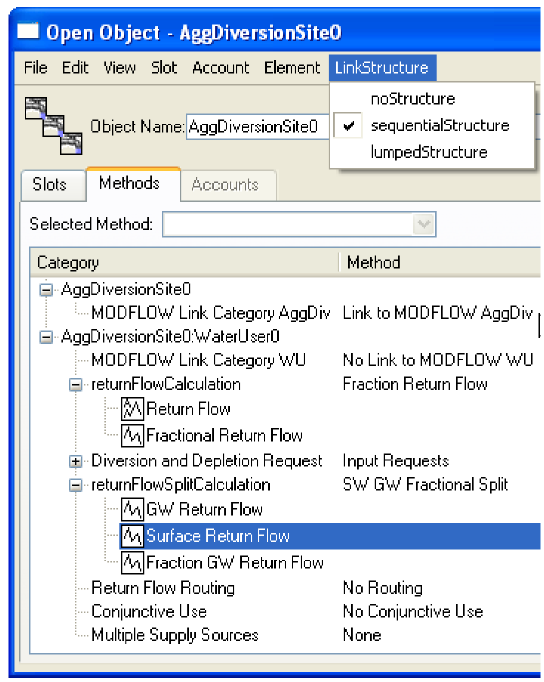
7.2.1.3.5 AggDiversion Site

Linking an AggDiversion Site object to MODFLOW is optional. The user may link none, some, or all the AggDiversion Site objects in a RiverWare model to MODFLOW.

Optional Data Exchange AggDiversion Site Configuration

A surface water body (e.g. riverside drain) could be represented in MODFLOW using the STR or SFR packages. When a surface water body is defined in this manner, it does not need to be explicitly represented in RiverWare. Flow between this type of MODFLOW surface water body and a RiverWare AggDiversion Site is optional. The user may choose to assign the surface return flow from a AggDiverson Site to a surface water body represented in MODFLOW (e.g. riverside drain).

Note: In the MODFLOW STR and SFR packages, stream/river/drain networks are assembled using reaches and segments. Reaches are joined together to form segments and all reachs in a segment share the same model properties. A reach can span up to one model cell, while segments can span multiple cells. Segments are numbered sequentially starting with the most upstream segment. The reaches in a segment are also numbered sequentially starting at the most upstream reach. Inflow into a MODFLOW stream network (STR or SFR) can only occur on the first reach in a segment, respectively (with one exception in the SFR package). This must be considered when the user matches corresponding variables between a RiverWare Water User and a MODFLOW segment. It is suggested that the user create tributary and diversion segments that are to be used only for data exchange with RiverWare.



- **Total Surface Return Flow**: It is optional to link the Total Surface Return Flow from all the Water User elements on an AggDiversion Site as inflow into a surface water body in MODFLOW (e.g. riverside drain). The user needs to select the "Sequential Structure" as the AggDiversion Site Link Structure and the "Link to MODFLOW WU" method in the "MODFLOW Link Category AggDiv" category, HERE (Section 3.1.4.2). The Total Surface Return Flow is calculated as the sum of the individual element's Surface Return Flow slot. The Total Surface Return Flow on the AggDiversion Site may be assigned as inflow into a MODFLOW STR or SFR segment using the computational subbasin structure. If no MODFLOW segment is assigned the user may link the total surface return flow as an inflow into another RiverWare object. The user may input a value for the Total Surface Return flow at the initial timestep if desired, otherwise an inflow value of zero will be transferred to MODFLOW for use in the first timestep.

7.2.1.4 Example

A simple example model is shown below. This example does not fully detail the setup of either a MODFLOW or RiverWare model. Instead it highlights the variables exchanged between the two model and shows how the user might match up MODFLOW cells with RiverWare objects.

Figure 2 shows a schematic of a RiverWare model. Figure 3 shows a schematic of how one RiverWare model matches up with two MODFLOW models and the variables that are passed between RiverWare and MODFLOW.

Figure 2. RiverWare Example Model

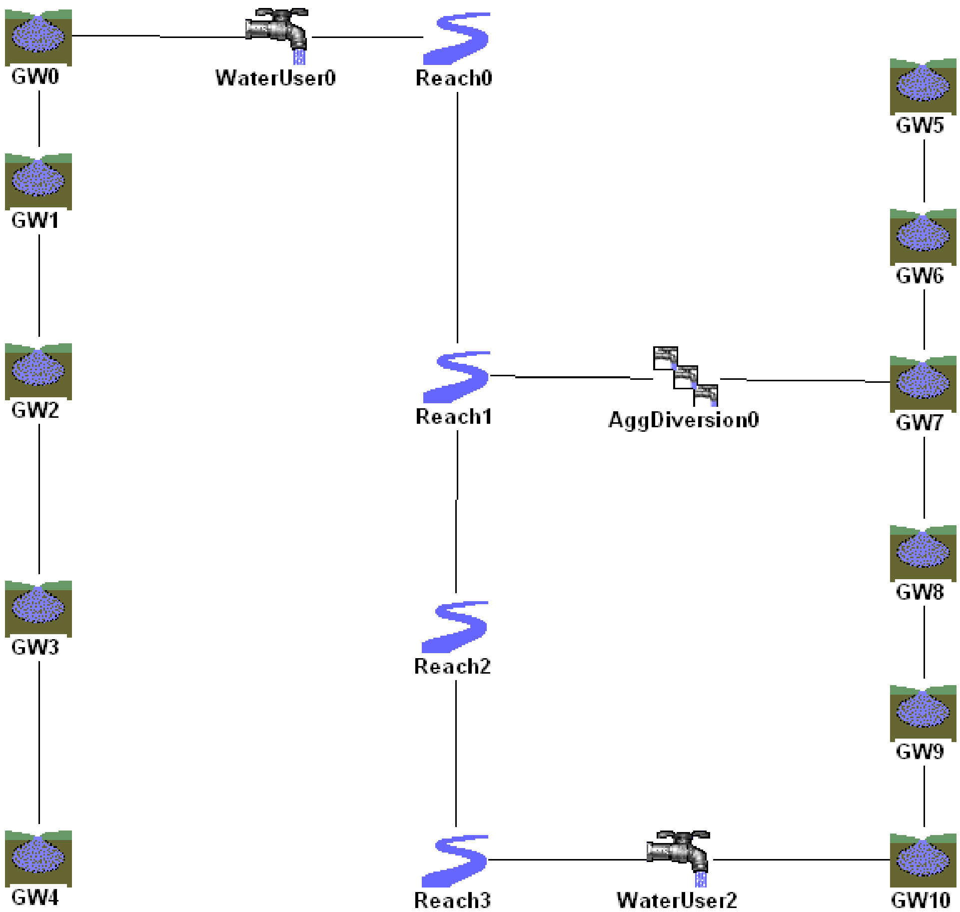


Figure 3. MODEL Interaction Overview: RiverWare Model with MODFLOW data exchanges shown

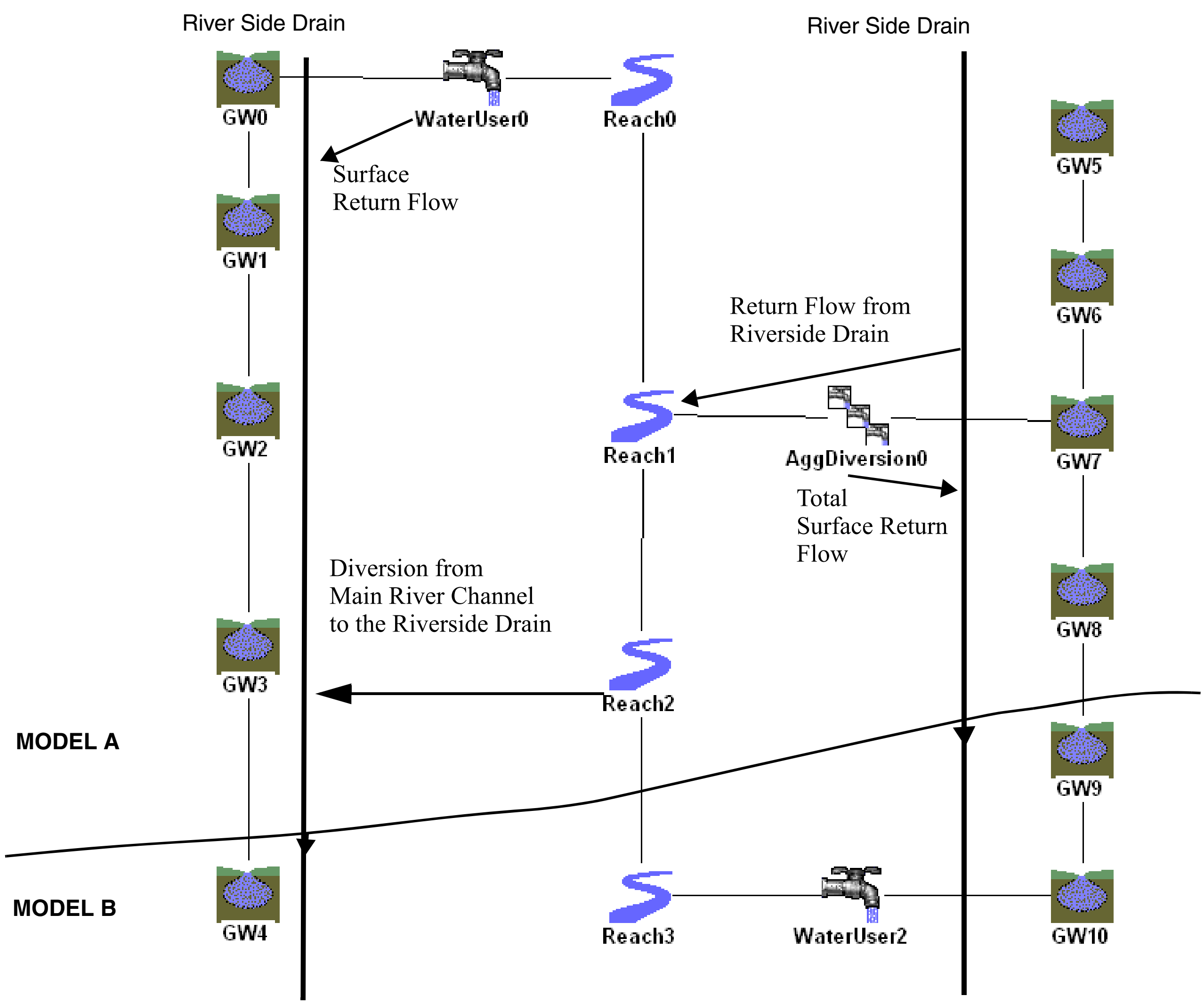
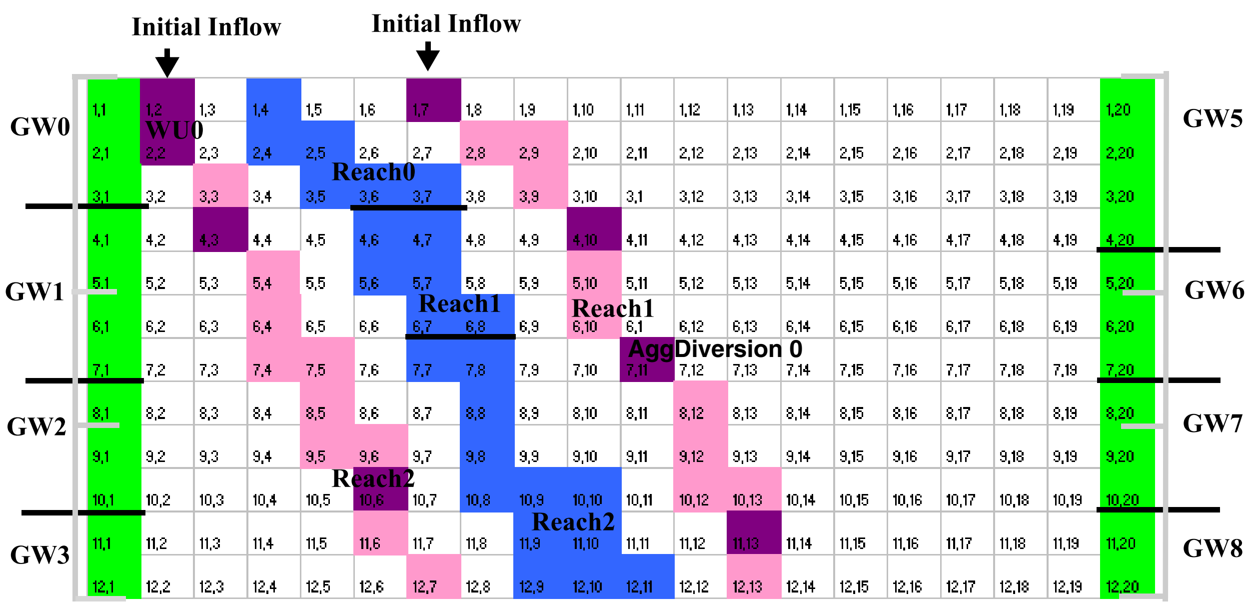


Figure 4 shows a schematic of how MODFLOW Model A matches up with the RiverWare model. In the MODFLOW model the blue cells represent RIV boundary conditions, green cells represent GHB boundary conditions, and the pink/purple cells indicate STR or SFR boundary conditions. The purple cells indicate the beginning a MODFLOW STR or SFR segment. The blue cells are partitioned into regions by black dividers, these regions represent all the cells corresponding to a given Reach and each region is labeled with the Reach name. The variables exchanged between the Reach and the blue cells are river Stage, as an elevation, and MF GainLoss (Table 3). For each RiverWare Reach, the Stage (interpolation) and MF GainLoss (summation) equations use the same MODFLOW cell to Reach correspondence. The division between GW objects is slightly more complicated than for the Reach, only one GW object is needed for summation of the "Lateral Flux from MODFLOW" from each cell, while two GW objects are needed to interpolate a GW head "Elevation" for each cell. The green cells (GHB) between the black dividers are summed to obtain the “Lateral Flux from MODFLOW” for the indicated RiverWare GW object. “Elevation” heads are interpolated for all green cells (GHB) between the gray dividers.

Figure 4. MODEL Interaction Overview: MODFLOW Grid with RiverWare object affiliations shown
Green = GHB - Model Boundary,
Blue = RIV - Main River Channel,
Pink/Purple = STR - Riverside Drains, Purple indicates the first reach in a segment the segment number is shown,
White = Cells not associated with the GHB, RIV, or STR packages - cells that will that will not communicate with RiverWare:



In Figure 3, flow between a surface water body in MODFLOW and a surface water body in RiverWare are noted. In Figure 4 the same flows are shown as corresponding to either the first (purple) or last reach in a MODFLOW segment.

All objects corresponding to MODFLOW Model A are grouped into a subbasin named ModelA, those associated with MODFLOW Model B are grouped into a subbasin named ModelB

- Subbasin name: ModelA
  - Reach 0
  - Reach 1
  - Reach 2
  - GW 0
  - GW 1
  - GW 2
  - GW 3
  - GW 5
- Subbasin name: ModelB
  - Reach 3
  - GW 4
  - GW 9
  - GW 10
  - Water User 2

- GW 6
- GW 7
- GW 8
- Water User 0
- AggDiversion 0

Table 3. Example **Reach Stage and GainLoss Map** slot - For use in the “Reach Stage”, “Weighted Interpolation” and “Reach Gain Loss”, “Summation” methods. This is used to assign RIV boundary cells to a Reach object. Stage and river GainLoss variables will be communicated between the two models. Inflow and Outflow Stage Weights are used in the interpolation equation. The user will input the layer, row, column, the inflow stage weight and outflow stage weight of each cell. The user will need to define the row heading as the appropriate Reach from one included on the subbasin.

	Layer	Row	Column	Inflow Stage Weight	Outflow Stage Weight
Reach0	1	1	4	0.8	0.2
Reach0	1	2	4	0.5	0.5
Reach0	1	2	5	0.5	0.5
Reach0	1	3	5	0.8	0.2
Reach0	1	3	6	0.8	0.2
Reach0	1	3	7	0.8	0.2
Reach1	1	4	6	0.7	0.3
Reach1	1	4	7	0.7	0.3
Reach1	1	5	6	0.5	0.5
Reach1	1	5	7	0.5	0.5
Reach1	1	6	7	0.3	0.7
Reach1	1	6	8	0.3	0.7
Reach2	1	7	7	0.9	0.1
Reach2	1	7	8	0.9	0.1
Reach2	1	8	8	0.8	0.2
Reach2	1	9	8	0.5	0.5
Reach2	1	10	8	0.3	0.7
Reach2	1	10	9	0.3	0.7
Reach2	1	10	10	0.3	0.7
Reach2	1	11	9	0.1	0.9
Reach2	1	11	10	0.1	0.9

Table 4. Example **GroundWater Elevation Upstream Map** slot. This is used in the "Groundwater Elevation", "Weighted Interpolation" method to map GHB boundary cells to a GW object. The upstream weight is used in the elevation interpolation equation. The user will input the layer, row, and column of each cell and the upstream elevation weight. The user will also need to define the row heading as the appropriate GW object from one included in the subbasin. This table should contain the same list of cells as the GroundWater Downstream Map

	Layer	Row	Column	Upstream Elevation Weight
GW 0	1	1	1	0.9
GW 0	1	2	1	0.7
GW 0	1	3	1	0.5
GW 0	1	4	1	0.3
GW 0	1	5	1	0.1
GW 1	1	6	1	0.7
GW 1	1	7	1	0.5
GW 1	1	8	1	0.3
GW 2	1	9	1	0.8
GW 2	1	10	1	0.6
GW 2	1	11	1	0.4
GW 2	1	12	1	0.2
GW 5	1	1	20	0.9
GW 5	1	2	20	0.7
GW 5	1	3	20	0.5
GW 5	1	4	20	0.3
GW 5	1	5	20	0.1
GW 6	1	6	20	0.7
GW 6	1	7	20	0.5
GW 6	1	8	20	0.3
GW 7	1	9	20	0.8
GW 7	1	10	20	0.6
GW 7	1	11	20	0.4
GW 7	1	12	20	0.2

Table 5. Example **GroundWater Elevation Downstream Map** slot. This is used in the “Groundwater Elevation”, “Weighted Interpolation” method to map GHB boundary cells to a GW object. The downstream weight is used in the elevation interpolation equation. The user will input the layer, row, and column of each cell and the downstream elevation weight. The user will also need to define the row heading as the appropriate GW object from one included in the subbasin. This table should contain the same list of cells as the GroundWater Upstream Map

	Layer	Row	Column	Downstream Elevation Weight
GW 1	1	1	1	0.1
GW 1	1	2	1	0.3
GW 1	1	3	1	0.5
GW 1	1	4	1	0.7
GW 1	1	5	1	0.9
GW 2	1	6	1	0.3
GW 2	1	7	1	0.5
GW 2	1	8	1	0.7
GW 3	1	9	1	0.2
GW3	1	10	1	0.4
GW3	1	11	1	0.6
GW 3	1	12	1	0.8
GW 6	1	1	20	0.1
GW 6	1	2	20	0.3
GW 6	1	3	20	0.5
GW 6	1	4	20	0.7
GW 6	1	5	20	0.9
GW 7	1	6	20	0.3
GW 7	1	7	20	0.5
GW 7	1	8	20	0.7
GW 8	1	9	20	0.2
GW 8	1	10	20	0.4
GW 8	1	11	20	0.6
GW 8	1	12	20	0.8

Table 6. Example of the **GroundWater Lateral Flux Map** slot for the "Groundwater Lateral Flux", "Summation" Method. This is used to map GHB boundary cells to a GW object. The user will input the layer, row, and column for each cell. The user will also need to assign the appropriate GW from one included on the subbasin as the row heading.

	Layer	Row	Column
GW 0	1	1	1
GW 0	1	2	1
GW 0	1	3	1
GW 1	1	4	1
GW 1	1	5	1
GW 1	1	6	1
GW 1	1	7	1
GW 2	1	8	1
GW 2	1	9	1
GW 3	1	10	1
GW 3	1	11	1
GW 3	1	12	1
GW 5	1	1	20
GW 5	1	2	20
GW 5	1	3	20
GW 5	1	4	20
GW 6	1	5	20
GW 6	1	6	20
GW 6	1	7	20
GW 7	1	8	20
GW 7	1	9	20
GW 8	1	10	20
GW 8	1	11	20
GW 8	1	12	20

Table 7. Additional slots that exchange data between RiverWare and MODFLOW

Simulation Object	Slot Value Received	MODFLOW Segment
Reach 2	Diversion	4
Water User 0	Surface Return Flow to MODFLOW	2
AggDiversion 0	Total Surface Return Flow	7
Reach 1	Local Inflow from MODFLOW	6

7.2.1.5 References

Harbaugh, A.W., Banta, E.R., Hill, M.C. and McDonald, M.G., 2000. MODFLOW-2000, The U.S. Geological Survey Modular Ground-Water Model - User Guide to Modularization Concepts and the Ground-Water Flow Process: U.S. Geological Survey Open File Report 00-92.

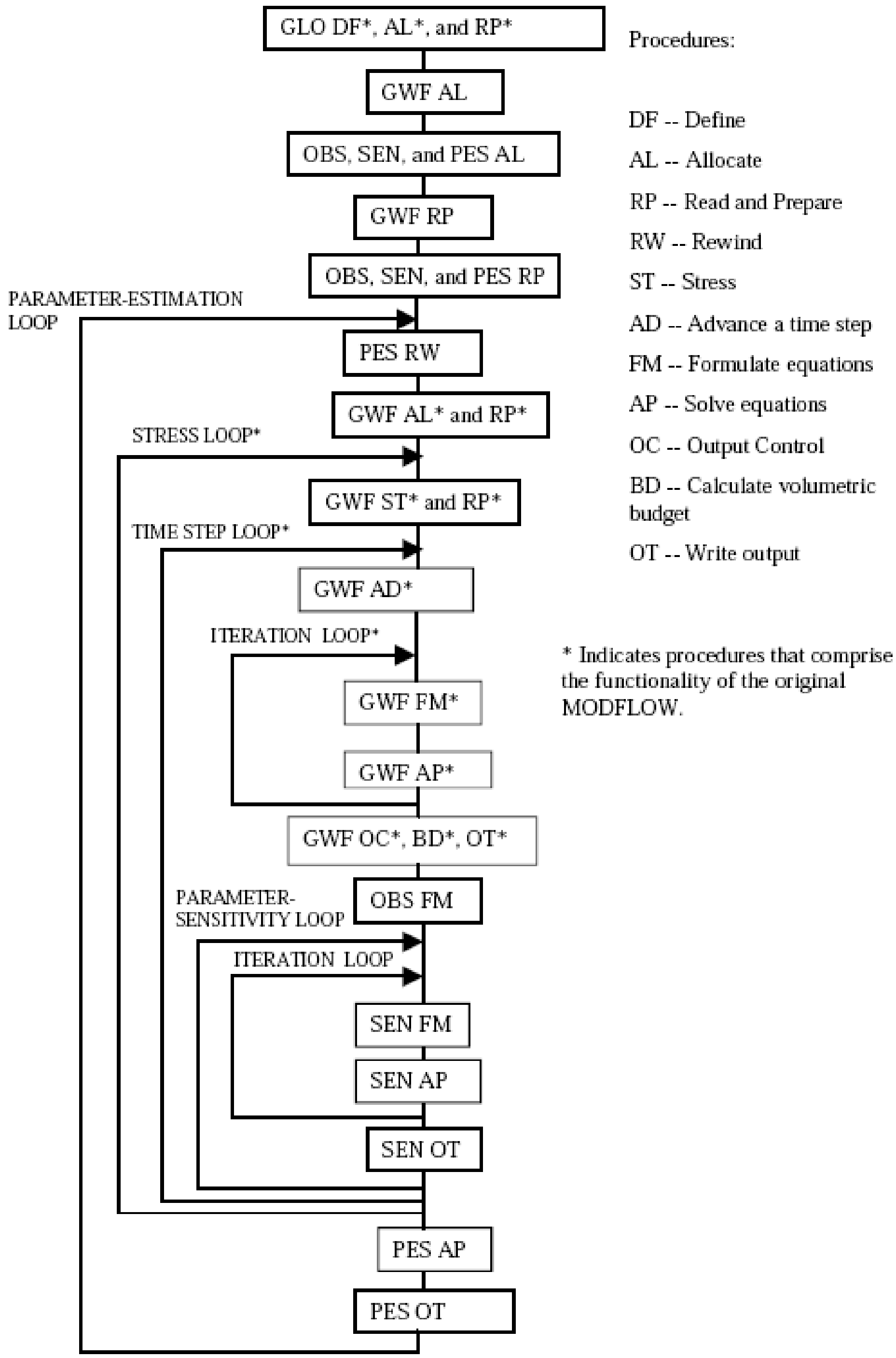
Maddock, T. and Baird, K.J, 2003. A Riparian Evapotranspiration Package for MODFLOW-96 and MODFLOW-2000. Department of Hydrology and Water Resources University of Arizona Research Laboratory for Riparian studies, University of Arizona, December 2003.

Prudic, D.E., Konikow, L.F., and Banta, E.R., 2004. A New Streamflow-Routing (SFR1) Package to Simulate Stream-Aquifer Interaction with MODFLOW-2000: U.S. Geological Survey Open File Report 2004-1042.

7.2.1.6 Appendix A: MODFLOW Package Description

Following is a description of the MODFLOW process from the MODFLOW 2000 documentation:

Figure 5. MODFLOW Control Flow (from MODFLOW 2000 documentation), Note, a Parameter-Estimation Loop is not performed in the RiverWare-MODFLOW library.



7.2.1.6.1 GHB and RIV Packages

The MODFLOW input files are similar for the GHB and RIV packages. Each cell that is designated in the GHB or RIV package is identified by Layer, Row, and Column. The Stage or the Bhead must be assigned in each time-step/peroid and for each qualifying cell. For example, the Stage parameter in MODFLOW corresponds to the value interpolated for a particular cell from the Inflow and Outflow Stages on the RiverWare Reach. An example of the input items lines in the GHB and RIV package that contain the variables of interest are shown below. The variable of interest is indicated in bold text (in this case the river stage and elevation). Note, data transferred from RiverWare to MODFLOW may not necessarily be directly input into the input file, here the input file configuration is shown to clearly identify the variable of interest.

MODFLOW Input:
GHB Package
Item 4b or Item 6 (Layer Row Column **Bhead** Cond(fact) [xyz])

RIV Package
Item 4b or Item 6 (Layer Row Column **Stage** Cond(fact) Rbot [xyz])

GHB Package Flux Equation

$$QB_n = CB_n \cdot (HB_n - h_{i,j,k})$$

n is a boundary number
QB_n is the flow into cell i,j,k from the boundary (L^3/T)
CB_n is the boundary conductance (L^2/T)
HB_n is the head elevation (Bhead) assigned to the external source (L)
$h_{i,j,k}$ is the flow into cell i,j,k from the boundary (L)

RIV Package Gain/Loss Equation

If $h_{i,j,k} > RBOT_n$ then

$$QRIV_n = CRIV_n \cdot (HRIV_n - h_{i,j,k})$$

If $h_{i,j,k} \le RBOT_n$ then

$$QRIV_n = CRIV_n \cdot (HRIV_n - RBOT_n)$$

n is the reach number
$QRIV_n$ is the flow between the river and the aquifer, taken as positive if it is directed into the aquifer (L^3/T)
$CRIV_n$ is the hydraulic conductance of the river-aquifer interconnection (L^2/T)
$HRIV_n$ is the water level elevation (Stage) in the river (L)
$RBOT_n$ is the river bottom elevation (L)
$h_{i,j,k}$ is the head in the cell underlying the river reach (L)

After the Bhead or Stage are assigned in each time-step/peroid and for each qualifying cell. The fluxes will be calculated by MODFLOW and transferred to RiverWare. The MODFLOW output files identify the fluxes (gain/loss and lateral flux) as a rate for each specified cell by layer, row, and column. An example of the of the standard formatted MODFLOW output file (.lst) is shown below. The variables of interest are indicated in bold text. Note, data from MODFLOW to be transferred to RiverWare or vice versa may not necessarily be taken directly from the Output file or input directly into the input file, however the configurations for these files shown to clear identify the variable of interests.

MODFLOW Output:
.lst File for GHB
    Head Dep Bounds Peroid # Step # (Titles included)
Boundary # Layer # Row # Col # **Rate #** (Titles included, one row for each boundary cell, Boundary number is sequential order from input file, the Boundary number is not included in the input File)

.lst File for Riv
    River Leakage Peroid # Step # (Titles included)
Reach # Layer # Row # Col # **Rate #** (Titles included, one row for each reach, Reach number is sequential ordering from input file, the Reach number is not included in the input File)

7.2.1.6.2 STR and SFR Packages

Inflow into a Segment

The STR or SFR packages in MODFLOW will be used to represent the riverside or interior drains. In these packages the surface return flow from the Water User must be specified as inflow into a segment. Inflow may only be specified into the first reach of a segment. There are two scenarios for inflow entering a segment: a single headwater inflow (there is no other source of flow into the segment) or multiple inflows (tributary inflow). The value entered for Flow (see input description below) in the STR and SFR packages differs slightly. Note the instructions below are meant to define what parameters in MODFLOW will receive a value from RiverWare and are not full input instructions for the STR and SFR packages.

In STR the value entered for Flow can be one of the following
1) headwater stream - Flow is the total inflow into the segment
2) stream with tributaries - Flow must be -1 to indicate that inflow into the segment is the outflow of one or more tributary segments
3) diversion stream - Flow is the total inflow into the segment.

In SFR the value entered for Flow can be one of the following
1) headwater stream - Flow is the total inflow into the segment
2) stream with tributaries - Flow is additional specified inflow or withdraw from the segment Note: This additional flow does not interact with the ground-water system in this segment.

3) diversion stream - value entered for Flow is not necessarily a flow quantity, the input changes depending on the diversion option chosen

For both STR and SFR, if the return flow enters a headwater stream segment then the surface return flow can be entered as the Flow value. For STR, if the return flow is in addition to another inflow then a tributary segment must be defined to handle the return flow. The surface return flow can then be entered as the Flow value in the tributary segment. For SFR, if the surface return flow is in addition to another inflow then the surface return flow can be entered as the Flow value in the segment or can be entered as the Flow value in a tributary segment. The surface return flow will be transferred from RiverWare to MODFLOW and input as Flow into the specified segment. The STR and SFR packages input configurations are different. The item/line numbers in the packages that contain the Flow value are shown below. The variable of interest (in this case surface return flow or surface water inflow) and the critical segment identification are indicated in bold text.

MODFLOW Input:
STR Package
Item 4b or Item 6: Layer Row Column **Seg** Reach **Flow** Stage Cond(fact) Sbot Stop
Note: Stage, Sbot, Stop are all elevations

SFR Package
Item 2 (KRCH IRCH JRCH **ISEG** IREACH RCHLEN)
Item 4b or Item 6a: **NSEG** ICALC OUTSEG IUPSEG {IPRIOR} {NSTRPTS} **FLOW** RUNOFF ETSW PPTSW {ROUGHCH} {ROUGHBK} {CDPTH} {FDPTH} {AWDTH} {BWDTH}
Note: Variables in {brackets} are not always specified in the input file
NSEG=segment #, NSEG is the same as ISEG which is entered in Item 2 and contains the layer, row, and column designation.

Diversion from a Segment

The return flow from a riverside or interior drain can be specified as a diversion in MODFLOW. No leakage can occur along a diversion segment. The diversion options available in each package are described below.

In STR, only one diversion option is available, a specified diversion. A segment must be declared as a diversion segment, the value entered as Flow (see STR input below) is the total flow diverted into the segment from the upstream segment. Note, if the Flow value in this segment is greater than the upstream segment outflow then no flow will be diverted. This diversion is removed from the last reach in the segment.

In SFR, there are several diversion options available. All diversions are taken out of the last reach in a segment, expect for the first option below. In the first option flow is diverted by specifying the a negative flow value for Flow in item 4b or 6a. In options 2 through 5 a segment must be declared as a diversion segment, the value entered for Flow changes depending on the option chosen. The diversion option is specified in MODFLOW using the IPRIOR flag.

- Specified diversion, flow diverted using this option is subtracted from the first reach in a segment.
- Specified diversion, flow diverted using this option is subtracted from the last reach in a segment and any remaining flow is routed as tributary inflow at the beginning of the next segment. This diversion option assumes that all flow in the segment can be diverted. IPRIOR = 0.
- Specified diversion, same as second option except, if the specified diversion is greater than flow in the segment then no flow will be diverted. IPRIOR=-1
- A specified fraction of the flow in the segment is diverted. IPRIOR=-2
- All excess flow beyond a given threshold is diverted. IPRIOR=-3

The item/line numbers corresponding to the input and output for the STR and SFR packages in MODFLOW for diversion flow are described below. An example excerpt from the MODFLOW input file and standard formatted output file (.lst) are shown below. Depending on which option is chosen the diversion may be easier to obtain from the input or the output file. For Option 1 the Flow value in item 4b or 6a is the amount diverted. For options 2-5 the value entered for Flow is not necessarily the amount diverted (for more detailed instructions on this value refer to USGS 2004-1042 p.44-45). The flow diverted to the segment can be read more easily from the output file as Flow Into the Stm Rch. The variable of interest (in this case diverted flow) and the critical segment identification are indicated in bold text. Note, data from MODFLOW to be transferred to RiverWare or vice versa may not necessarily be taken directly from the Output file or input directly into the input file, however the configurations for these files shown to clear identify the variable of interests.

MODFLOW Input:
STR Package
Item 4b or Item 6: Layer Row Column **Seg** Reach **Flow** Stage Cond Sbot Stop
Note: Stage, Sbot, Stop are all elevations

SFR Package
Item 2 (KRCH IRCH JRCH **ISEG** IREACH RCHLEN)
Item 4b or Item 6a: **NSEG** ICALC OUTSEG IUPSEG {IPRIOR} {NSTRPTS} **FLOW** RUNOFF ETSW PPTSW {ROUGHCH} {ROUGHBK} {CDPTH} {FDPTH} {AWDTH} {BWDTH}
Note: Variables in {brackets} are not always specified in the input file
NSEG=segment #, NSEG is the same as ISEG which is entered in Item 2 and contains the layer, row, and column designation.

MODFLOW Output:
.lst File for STR
Stream Flow Out, Time Step #, Stress Peroid #
Layer, Row, Column, **Stream-Number,** Reach Number, **Flow Into Stream Reach**, Flow Into Aquifer, Flow out of Stream Reach, Head in Stream (Column Headings)

.lst File for SFR

Stream Leakage Peroid # Step #

Layer, Row, Col, **Stream Seg. No.,** Rch Number, **Flow Into Strm. Rch.**, Flow Into Aquifer, Flow out of Stm. Rch., Ovrlnd. Runoff, Direct Precip, Stream ET, Stream Head, Stream Depth, Stream Width, Streambed Conductnc., Streambed Gradient (Column Headings)

8. Confluence

This object models a confluence of water.

General Slots

☞ INFLOW1

Type: AggSeriesSlot

Units: FLOW

Description: flow from one side into the confluence

Information:

I/O: Optional; can input, set by propagation across a link, or calculated

Links: May be linked to the Outflow of any object.

☞ INFLOW2

Type: AggSeriesSlot

Units: FLOW

Description: flow from the other side into the confluence

Information:

I/O: Optional; can be input, set by propagation across a link, or calculated

Links: May be linked to the Outflow of any object.

☞ OUTFLOW

Type: AggSeriesSlot

Units: FLOW

Description: flow out of the confluence

Information:

I/O: Optional; can be input, set by propagation across a link, or calculated

Links: May be linked to the Inflow of any object.

8.1 User Methods

8.1.1 Confluence Solution Direction

This method category is used to specify the direction the Confluence solves (downstream given the two inflows or upstream given the outflow and one inflow). For basic Simulation runs this does not make a difference and the default method should remain selected. However, in Rulebased Simulation the user may need to limit the Confluence to downstream solution only.

8.1.1.1 Solve Upstream or Downstream

This is the default method and should remain selected for basic Simulation. There are no slots or calculations specifically associated with this method. It simply allows all dispatch methods to be active.

8.1.1.2 Solve Downstream Only

This method may need to be selected in Rulebased Simulations where the Confluence can solve in the downstream direction only. When this method is selected, only the solveOutflow dispatch method is selected.

8.2 Dispatch Methods

This object solves for inflow1, inflow2, or outflow, when the other two are known. If the Solve Upstream or Downstream method is selected, then all dispatch methods are active. However, if Solve Downstream Only is selected, then only the solveOutflow method is active.

8.2.1 solveOutflow

Solves for outflow.

Required Knowns

- ☞ **Inflow1**
- ☞ **Inflow2**

Required Unknowns

- ☞ **Outflow**

The equation for this dispatch method is:

$$Outflow = Inflow1 + Inflow2$$

8.2.2 solveInflow1

Solves for Inflow1.

Required Knowns

- ☞ **Outflow**
- ☞ **Inflow2**

Required Unknowns

- ☞ **Inflow1**

The equation for this dispatch method is:

$$Inflow1 = Outflow - Inflow2$$

8.2.3 solveInflow2

Solves for Inflow2.

Required Knowns

☞ **Inflow1**

☞ **Outflow**

Required Unknowns

☞ **Inflow2**

The equation for this dispatch method is:

$$Inflow2 = Outflow - Inflow1$$

9. Control Point

The Control Point object models a location on a river where certain flow requirements (maximum and minimums) must be maintained. Oftentimes, the control points are used for flood control operations such that the flood control releases do not cause an exceedence of channel capacity at these points.

General Slots

☞ Inflow

Type: SeriesSlot

Units: FLOW

Description: Flow rate at the entrance of the object

Information:

I/O: Optional; can be input or linked

Links: May be linked to the Outflow of another object.

☞ Outflow

Type: SeriesSlot

Units: FLOW

Description: Flow rate at the exit of the object

Information:

I/O: Output only

Links: May be linked to the Inflow of another object

☞ Forecast Period

Type: TableSlot

Units: NONE

Description: The forecast period is a number of timesteps, including the current simulation timestep, that is used in the algorithms for calculating forecasted hydrology, regulation discharge and flood releases. This can be propagated from a computational subbasin of which the control point is a member.

Information:

I/O: Required input

Links: Not linkable

☞ UPSTREAM RESERVOIRS

Type: ListSlot

Units: NONE

Description: List of upstream reservoirs that route to this control point during the forecast period.

Information: Routing coefficients must be provided in the Routing Coefficients slot for each reservoir in this list.

I/O: Required Input

Links: Not linkable

☞ ROUTING COEFFICIENTS

Type: TableSlot

Units: NONE, NONE

Description: Contains linear routing coefficients for routing flow from each reservoir specified in the Upstream Reservoirs slot.

Information: Each column corresponds to a reservoir specified in the Upstream Reservoirs slot. Each row represents a timestep starting with the current timestep. At each intersection of a row and a column is a decimal number to represent the routing coefficient from that reservoir for that timestep. The coefficients in a column for a reservoir must sum to 1.0 and must not be NaN. NOTE: the routing coefficients are not used in the solution of Control Point Outflow. They are strictly used in flood control and other basin wide operations **HERE (Section 9.1.8)**.

I/O: Required Input if Upstream Reservoirs are defined

Links: Not Linkable

9.1 User Methods

9.1.1 Local Inflow

The Local Inflow methods determine how the Local Inflow is calculated.

9.1.1.1 None

This method is the default for the category, and should be selected when modeling of local inflow is not desired. There are no slots specifically associated with this method.

9.1.1.2 Input Local Inflow

This method treats Local Inflow as an optional input.

SLOTS SPECIFIC TO THIS METHOD

☞ **LOCAL INFLOW**

Type: SeriesSlot

Units: FLOW

Description: The incremental local inflow to the control point

Information: If a disaggregation method is selected, the local inflow will be set to the result of the last stage of these methods.

I/O: Output

Links: Usually not linked

9.1.1.3 Forecast Local Inflows

The selection of this method activates the category Generate Forecast Inflows which contains methods for forecasting the local inflow.

9.1.2 Generate Forecast Inflows

9.1.2.1 None

This method is the default for this category which does nothing.

9.1.2.2 Geometric Recession

On each timestep in the forecast period, this method will adjust the inflow hydrographs. If the Compute Forecast Period Incremental Local Inflows method is selected the Cumulative Local Inflow is used to forecast and set the Forecasted Cumulative Local Inflow. If the Compute Forecast Period Incremental Local Inflows methods is not selected, the input Deterministic Incremental Local Inflow slot is used to forecast and set the Local Inflow series slot.

Slots Specific to this Method

☞ Deterministic Incremental Local Inflow

Type: SeriesSlot

Units: FLOW

Description: This slots holds a timeseries of the actual Local Inflows to the control point. These values are then adjusted by the forecast method and set on the Local Inflow slot.

Information: At the end of the run, the Local Inflow slot will be identical to this slot. If the Compute Full Run Incremental Local Inflows method is selected, values from the Incremental Local Inflow slot will be copied into this slot prior to the forecast. If the Compute Forecast Period Incremental Inflows method is selected, this slot is not used (i.e. inputs are ignored).

Information:

I/O: Input only

Links: Not linkable

☞ Local Inflow

Type: SeriesSlot

Units: FLOW

Description: The incremental local inflow to the control point

Information: If a disaggregation method is selected, the local inflow will be set to the result of the last stage of these methods.

I/O: Output

Links: Usually not linked

☞ Period of Perfect Knowledge

Type: ScalarSlot

Units: NONE

Description: Number of timesteps for which the forecast will equal the Deterministic Incremental Local Inflow, i.e., the forecast is known with complete certainty.

Information: Minimum value of 1; maximum value equal to the number of timesteps in the forecast period.

I/O: Input only

Type: Not linkable

☞ Recession Factor

Type: ScalarSlot

Units: NONE

Description: A decimal value that is multiplied by the previous Local Inflow to determine the current value after the Period of Perfect Knowledge.

Information:

I/O: Input only

Links: Not linkable

This method uses different slots and sets different slots if the Compute Forecast Period Incremental Local Inflows is selected. But, there is always a Source slot and a Target slot where the Source slot is input and used to forecasted and set the Target slot. If the Compute Forecast Period Incremental Local Inflows is selected, the computational subbasin calls this method and uses the Cumulative Local Inflow (Source) to forecast and set the Forecasted Cumulative Local Inflow (Target). If it is not selected, the reservoir calls this method at the beginning of the timestep and uses the Deterministic Incremental Local Inflow (Source) slot to forecast and set the Local Inflow slot (Target). The following uses the Source/Target terminology to describe the methods.

The Source slot values are required inputs for each timestep. At the beginning of each controller timestep, the Geometric Recession method is executed. For each forecast timestep within the period of perfect knowledge, the Target is set to the Source value. For each forecast timestep after the period of perfect knowledge, the Target is set by multiplying the value of the Target from the previous timestep by the constant recession factor.

A value for the Source slot must be known at every timestep during the run. If Target values are desired past the end of the run, there must also be values in the Source slot at timesteps past the end of the run. If values for Source slot are not entered past the end of the run, the Target values for these timesteps are assumed to be zero.

9.1.2.3 Exponential Recession

On each timestep in the forecast period, this method will adjust the inflow hydrographs. If the Compute Forecast Period Incremental Local Inflows method is selected the Cumulative Local Inflow is used to forecast and set the Forecasted Cumulative Local Inflow. If the Compute Forecast Period Incremental Local Inflows methods is not selected, the input Deterministic Incremental Local Inflow slot is used to forecast and set the Local Inflow series slot.

SLOTS SPECIFIC TO THIS METHOD

☞ DETERMINISTIC INCREMENTAL LOCAL INFLOW

Type: SeriesSlot

Units: FLOW

Description: This slots holds a timeseries of the actual Local Inflows to the control point. These values are then adjusted by the forecast method and set on the Local Inflow slot.

Information: At the end of the run, the Local Inflow slot will be identical to this slot. If the Compute Full Run Incremental Local Inflows method is selected, values from the Incremental Local Inflow slot will be copied into this slot prior to the forecast. If the Compute Forecast Period Incremental Inflows method is selected, this slot is not used (i.e. inputs are ignored).

I/O: Input only

Links: Not linkable

☞ LOCAL INFLOW

Type: SeriesSlot

Units: FLOW

Description: The incremental local inflow to the control point

Information: If a disaggregation method is selected, the local inflow will be set to the result of the last stage of these methods.

I/O: Output

Links: Usually not linked

☞ PERIOD OF PERFECT KNOWLEDGE

Type: ScalarSlot

Units: NONE

Description: Number of timesteps for which the forecast will equal the Deterministic Incremental Local Inflow, i.e., the forecast is known with complete certainty.

Information: Minimum value of 1; maximum value equal to the number of timesteps in the forecast period.

I/O: Input only

Links: Not linkable

☞ Minimum Forecasted Flow

Type: SeriesSlot

Units: FLOW

Description: The minimum forecasted flow.

Information: If the computed value for Local Inflow is less than the Minimum Forecasted Flow, it is set to the Minimum Forecasted Flow.

I/O: Input only

Links: Not linkable

☞ Low Flow Threshold

Type: ScalarSlot

Units: FLOW

Description: The flow rate that dictates whether to use the Low Flow Recession Coefficient or the High Flow Recession Coefficient.

Information:

I/O: Input only

Links: Not linkable

☞ Low Flow Recession Coefficient

Type: ScalarSlot

Units: NONE

Description: The recession coefficient used when the Deterministic Incremental Local Inflow (at the end of the Period of Perfect Knowledge) is below or equal to the Low Flow Threshold.

Information:

I/O: Input only

Links: Not linkable

☞ High Flow Recession Coefficient

Type: ScalarSlot

Units: NONE

Description: The recession coefficient used when the Deterministic Incremental Local Inflow (at the end of the Period of Perfect Knowledge) is above the Low Flow Threshold.

Information:

I/O: Input only

Links: Not linkable

This method uses different slots and sets different slots if the Compute Forecast Period Incremental Local Inflows is selected. But, there is always a Source slot and a Target slot where the Source slot is input and used to forecasted and set the Target slot. If the Compute Forecast Period Incremental Local Inflows is selected, the computational subbasin calls this method and uses the Cumulative Local Inflow (Source) to forecast and set the Forecasted Cumulative Local Inflow (Target). If it is not selected, the reservoir calls this method at the beginning of the timestep and uses the Deterministic Incremental Local Inflow (Source) slot to forecast and set the Local Inflow slot (Target). The following uses the Source/Target terminology to describe the methods.

The Source slot values are input for each timestep. At the beginning of each controller timestep, the Exponential Recession method is executed. For each forecast timestep within the period of perfect knowledge, the Target is set to the Source value. For each forecast timestep after the period of perfect knowledge, the Target slot is set as described below:

$$ForecastedFlow = MAX\left[MinimumForecastedFlow, \left(Source \bullet e^{\frac{(-C)t}{T}}\right)\right]$$

where Source is the value in the Source slot at the end of the period of perfect knowledge, C is the recession coefficient, t is the elapsed time of the forecast period, and T is the total time from the end of the period of perfect knowledge to the end of the forecast period.

If the Source at the end of the period of perfect knowledge is negative, the Target at that timestep is exactly equal to the Source. However, the Source used in the recession equation, is the last positive value for Source. In the event that there is not a positive value for the Source, RiverWare issues a warning, and all values for Local Inflow within the forecast period will be set to the Minimum Forecasted Flow.

A value for the Source slot must be known at every timestep during the run. If the Target values are desired past the end of the run, there must also be values in the Source slot at timesteps past the end of the run. If values for Source are not entered past the end of the run, the Target values for these timesteps are assumed to be zero.

9.1.3 Local Inflow Spatial Disaggregation on Subbasin

The Local Inflow Spatial Disaggregation on Subbasin category allows the user to include the Control Point in a spatial disaggregation of local inflow. The methods in this category are executed from the subbasin containing this control point; click **HERE (Section 7.1.20)** for more information.

9.1.3.1 None

This method is the default for the Local Inflow Spatial Disaggregation on Subbasin category and should be selected when spatial disaggregation of local inflows is not desired. There are no slots specifically associated with this method.

9.1.3.2 WAM Precipitation Curve Number

This method holds the slots necessary for the calculation of Distributed Flow at an un-gaged control point using the known flow at the Gage Control Point. This method uses the procedure used in the Water Availability Models (WAM) developed by Dr. Ralph Wurbs at Texas A&M University. There are eight slots associated with this method: Local Inflow, Distributed Flow, Drainage Area, Curve Number, Mean Precipitation, Upstream Gages, Downstream Gage, and Excluded Gages. These slots are accessed by the computational subbasin when executing the computational subbasin’s WAM Precipitation Curve Number method. The computational subbasin uses the input Distributed Flow values in the Gage Control Point to calculate and set the Distributed Flow values at all other non-gage control points in the subbasin. Refer to the computational subbasin’s WAM Precipitation Curve Number documentation for details, **HERE (Section 7.1.20.2)**.

Slots Specific to this Method

☞ Local Inflow

Type: SeriesSlot

Units: FLOW

Description: The disaggregated local inflow to the control point

Information: If a disaggregation or forecast method is selected, the local inflow will be set to the result of the last stage of these methods.

I/O: Output

Links: Usually not linked

☞ Distributed Flow

Type: SeriesSlot

Units: FLOW

Description: The cumulative inflow to the control point

Information: Accessed or set in the Spatial Disaggregation method on the computational subbasin. The default timestep size is monthly, but can be changed by the user. Timestep size is independent of model timestep size.

I/O: Input or computed. If input, the control point should be a Gage Control Point. Computed for all non-gage control points and set with an input flag.

Links: Usually not linked

☞ Drainage Area

Type: ScalarSlot

Units: AREA

Description: The total drainage area contributing to the control point

Information: Used in the Spatial Disaggregation method on the computational subbasin to calculate Distributed Flow

I/O: Required Input

Links: Not Linkable

☞ Curve Number

Type: ScalarSlot

Units: NO UNITS

Description: The NRCS curve number for the subbasin contributing to the control point

Information: Used in the Spatial Disaggregation method on the computational subbasin to calculate Distributed Flow. It must be between 0 and 100.

I/O: Required Input

Links: Not linkable

☞ Mean Precipitation

Type: ScalarSlot

Units: LENGTH

Description: The monthly mean precipitation for the subbasin contributing to the control point. It must be greater than 0.

Information: Used in the Spatial Disaggregation method on the computational subbasin to calculate Distributed Flow.

I/O: Required Input

Links: Usually not linked

☞ Upstream Gages

Type: ListSlot

Units: NONE

Description: The list of upstream gages for this control point or gage

Information: Used in the Spatial Disaggregation method on the computation subbasin. The calculations require upstream gages to be entered for the downstream gage and for each Un-gaged Control Point.

I/O: Input

Links: Not Linkable

☞ Downstream Gage

Type: ListSlot

Units: NONE

Description: The downstream gage for this control point

Information: Used in the Spatial Disaggregation method on the computational subbasin. The calculations require that a downstream gage be entered for all Un-gaged Control Points.

I/O: Input

Links: Not Linkable

☞ Excluded Gages

Type: ListSlot

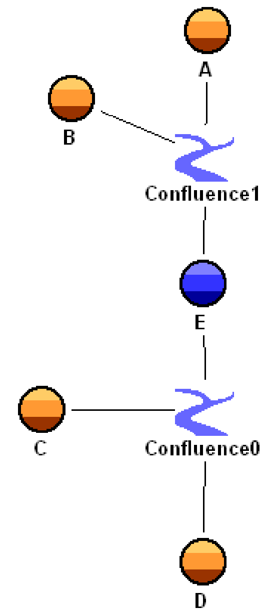
Units: NONE

Description: The list of gages excluded from the gain calculation for the downstream gage control point

Information: Used in the Spatial Disaggregation method on the computational subbasin. This list is set on un-gaged control points using the WAM Curve Number method or the Drainage Area method. For most un-gaged control points, this list is empty. When the list is used, it specifies which gages should be lumped into the gain for its downstream gage control point. Include only gages that are tributary to the downstream gage but are not tributary to the un-gaged control point. For example, un-gaged control point E excludes gaged control point C in its calculation.

I/O: Input

Links: Not Linkable



9.1.3.3 Drainage Area

This method holds the slots necessary for the calculation of Distributed Flow at an Un-gaged Control Point using the known flow at the Gage Control Point(s). This method uses a ratio of un-gaged drainage area to gaged drainage area to scale gage flows. There are six slots associated with this method: Local Inflow, Distributed Flow, Drainage Area, Upstream Gages, Downstream Gage, and Excluded Gages. These slots are accessed by the computational subbasin when executing the computational subbasin's Drainage Area method. The computational subbasin uses the input Distributed Flow values in the Gage Control Point(s) to calculate and set the Distributed Flow values at all other non-gage control points in the subbasin. Refer to the computational subbasin's Drainage Area documentation for details, HERE (Section 7.1.20.3).

SLOTS SPECIFIC TO THIS METHOD

☞ LOCAL INFLOW

Type: SeriesSlot

Units: FLOW

Description: The disaggregated local inflow to the control point

Information: If a disaggregation or forecast method is selected, the local inflow will be set to the result of the last stage of these methods.

I/O: Output

Links: Usually not linked

☞ Distributed Flow

Type: SeriesSlot

Units: FLOW

Description: The cumulative inflow to the control point

Information: Accessed or set in the Spatial Disaggregation method on the computational subbasin. The default timestep size is monthly, but can be changed by the user. Timestep size is independent of model timestep size.

I/O: Input or computed. If input, the control point should be a Gage Control Point. Computed for all non-gage control points and set with an input flag.

Links: Usually not linked

☞ Drainage Area

Type: ScalarSlot

Units: AREA

Description: The total drainage area contributing to the control point

Information: Used in the Spatial Disaggregation method on the computational subbasin to calculate Distributed Flow

I/O: Required Input

Links: Not Linkable

☞ Upstream Gages

Type: ListSlot

Units: NONE

Description: The list of upstream gages for this control point or gage

Information: Used in the Spatial Disaggregation method on the computation subbasin. The calculations require upstream gages to be entered for the downstream gage and for each Un-gaged Control Point.

I/O: Input

Links: Not Linkable

☞ Downstream Gage

Type: ListSlot

Units: NONE

Description: The downstream gage for this control point

Information: Used in the Spatial Disaggregation method on the computational subbasin. The calculations require that a downstream gage be entered for all Un-gaged Control Points.

I/O: Input

Links: Not Linkable

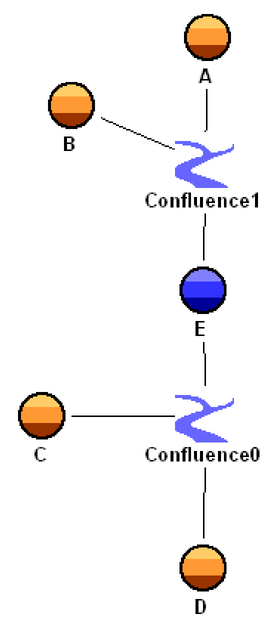
☞ Excluded Gages

Type: ListSlot

Units: NONE

Description: The list of gages excluded from the gain calculation for the downstream gage control point

Information: Used in the Spatial Disaggregation method on the computational subbasin. This list is set on un-gaged control points using the WAM Curve Number method or the Drainage Area method. For most un-gaged control points, this list is empty. When the list is used, it specifies which gages should be lumped into the gain for its downstream gage control point. Include only gages that are tributary to the downstream gage but are not tributary to the un-gaged control point. For example, un-gaged control point E excludes gaged control point C in its calculation.



I/O: Input

Links: Not Linkable

9.1.4 Gage Control Point category

A new category, Gage Control Point, is added to the Control Point when the WAM Precipitation Curve Number or Drainage Area method is selected. Methods in this category are used to indicate which Control Points are the “gage” control points. There are two methods in this category: the default, no action None method and the Gage Control Point method

9.1.4.1 None

This is the default, no-action method. It indicates this Control Point is not a “gage” control point.

9.1.4.2 Gage Control Point method

When selected, the Gage Control Point method indicates that the control point is considered a “gage” control point in the WAM Precipitation Curve Number or Drainage Area method. When this method is selected, the Control Point changes from a blue color to an orange color.

9.1.5 Local Inflow Temporal Disaggregation on Subbasin

The Local Inflow Temporal Disaggregation on Subbasin category allows the user to specify that this Control Point should be part of a Local Inflow Temporal Disaggregation as called from the containing subbasin. Click HERE (Section 7.1.21) for more information.

9.1.5.1 None

This method is the default for the Local Inflow Temporal Disaggregation on Subbasin category and should be selected when temporal disaggregation of local inflows is not desired. There are no slots specifically associated with this method.

9.1.5.2 Specified Factors

This method holds the control point slots necessary for the execution of the Specified Factors method on the computational subbasin. The computational subbasin temporally disaggregates local inflows by multiplying the Distributed Flow values on the control point by the Temp Disagg Factors values on the computational subbasin and sets the Temporally Disaggregated Flow values as input on the control point. Refer to the computational subbasin's Specified Factors documentation for details. The Specified Factors method on the control point performs no actions.

Slots Specific to this Method

☞ **Local Inflow**

Type: SeriesSlot

Units: FLOW

Description: The disaggregated local inflow to the control point

Information: If a disaggregation or forecast method is selected, the local inflow will be set to the result of the last stage of these methods.

I/O: Output

Links: Usually not linked

☞ **Distributed Flow**

Type: SeriesSlot

Units: FLOW

Description: The cumulative local inflow to the control point

Information: Accessed by the Temporal Disaggregation method on the computational subbasin. The default timestep size is monthly, but can be changed by the user. Timestep size is independent of model timestep size.

I/O: Required input.

Links: Usually not linked

☞ Temporally Disaggregated Flow

Type: SeriesSlot

Units: FLOW

Description: The temporally disaggregated local inflow to the control point

Information: Set by the Temporal Disaggregation method on the computational subbasin.

I/O: Computed and set with the input flag.

Links: Usually not linked

9.1.6 Compute Incremental Local Inflows on Subbasin

The Compute Incremental Local Inflows on Subbasin category is used to specify whether the Control Point contains cumulative local inflows that must be disaggregated. Methods allow the user to either disaggregate all cumulative flows at once or forecast the cumulatives then disaggregate. The disaggregation is executed from the Computational Subbasin containing this Control Point; click HERE (Section 7.1.22) for more information.

The Compute Incremental Local Inflows on Subbasin category contains three methods, None, Compute Full Run Incremental Local Inflows and Compute Forecast Period Incremental Local Inflows methods.

When the Input Local Inflows is selected in the Local Inflow Calc category, the Compute Full Run Incremental Local Inflows becomes available. The new method contains two new slots: Cumulative Local Inflow and Incremental Local Inflow. Data must be input into the Cumulative Local Inflow slots. The computational subbasin will execute the Compute Full Run Incremental Local Inflows method and set the Incremental Local Inflow slots as input. Setting the Incremental Local Inflow slots as input prevents the slot values from being cleared in future model runs when the subbasin is disabled. The computational subbasin will first check that Compute Full Run Incremental Local Inflows method has been selected on all reservoirs and control points in the basin. During the model run, the Compute Full Run Incremental Local Inflows method will copy the value in the Incremental Local Inflow slot and set the Local Inflow slot.

When a forecast method is selected in the Generate Forecast Inflows methods, a new method becomes available within the Compute Incremental Local Inflows on Subbasin category: Compute Forecast Period Incremental Local Inflows. This method contains the Cumulative Local Inflow slot and the Forecasted Cumulative Local Inflow slot. When this method is selected the computational subbasin executes the Compute Forecast Period Incremental Local Inflows method and sets the Local Inflow slots.

9.1.6.1 None

This method is the default for the Compute Incremental Local Inflows on Subbasin category and should be selected when local inflow data is not cumulative or the computation of incremental local inflows is not desired. There are no slots specifically associated with this method.

9.1.6.2 Compute Full Run Incremental Local Inflows

The Compute Full Run Incremental Local Inflows method is available from the Compute Incremental Local Inflows on Subbasin category on the Control Point. This method is only available if the Input Local Inflows method is selected in the Local Inflow category. The method contains two slots: Cumulative Local Inflow and Incremental Local Inflow. These slots will be accessed by the computational subbasin when executing the Compute Full Run Incremental Local Inflows method. The computational subbasin will use the user input

Cumulative Local Inflow value to calculate and set the Incremental Local Inflow slot. The Compute Full Run Incremental Local Inflows method on the control point then copies the results of the calculation in the Incremental Local Inflow slot values to the Local Inflow series slot.

Slots Specific to this Method

☞ Cumulative Local Inflow

Type: SeriesSlot

Units: Flow

Description: The cumulative local inflow to the control point

Information: Local inflow is cumulative either (1) between headwater control points and the first reservoir in the river system and also throughout the system between two reservoirs, or (2) throughout the entire river system.

I/O: Required Input

Links: Not linkable

☞ Incremental Local Inflow

Type: SeriesSlot

Units: Flow

Description: The incremental local inflow to the control point

Information: This slot is set by the Calculate Incremental Flows method on the computational subbasin and represents the actual local inflow to the control point. At the end of each timestep, the Local Inflow slot will be identical to this slot.

I/O: Computed and set with the Input flag

Links: Usually not linked

9.1.6.3 Compute Forecast Period Incremental Local Inflows

The Compute Forecast Period Incremental Local Inflows method is available from the Compute Incremental Local Inflows on Subbasin category on the Control Point only if the Forecast Local Inflows method is selected in the Local Inflow category and one of the forecasting methods (i.e. Geometric Recession or Exponential Recession) is selected in the Generate Forecast Inflows category. If this method is selected, but the control point is not part of a subbasin with the appropriate methods selected, an error will be issued. The method contains two slots: Cumulative Local Inflow and Forecasted Cumulative Local Inflow. These slots will be accessed by the computational subbasin when executing the Compute Forecast Period Incremental Local Inflows method. The computational subbasin will use the user input Cumulative Local Inflow value to forecast and set the Forecasted Cumulative Local Inflow slot.

This is further described HERE (Section 7.1.22.3) and for USACE-SWD models, HERE (USACE_SWD.pdf, Section 2.8).

Slots Specific to this Method

☞ Cumulative Local Inflow

Type: SeriesSlot

Units: FLOW

Description: The cumulative local inflow to the control point

Information: Local inflow is cumulative either (1) between headwater control points and the first reservoir in the river system and also throughout the system between two reservoirs, or (2) throughout the entire river system.

I/O: Required Input

Links: Not linkable

☞ Forecasted Cumulative Local Inflow

Type: SeriesSlot

Units: FLOW

Description: The forecasted cumulative local inflow to the control point.

Information: This slot is set by the selected Generate Forecast Local Inflows method on the control point as called from the computational subbasin. It represents the cumulative local inflow to the control point forecasted throughout the forecast period. This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).

I/O: Output only

Links: Not linked

9.1.7 Include Locals in Outflow

9.1.7.1 Locals Included in Outflow

This method is the default for this Category and does not make any changes to the operation of the model. Local inflows will be included in the outflow.

9.1.7.2 Locals Not Included in Outflow

This method will prevent local inflows from being included in the Outflow slot. A new slot, Total Discharge, will be created that accounts for the actual flow at the control point including upstream inflows and local inflows. This method will set the Outflow slot equal to inflow.

SLOTS SPECIFIC TO THIS METHOD

☞ TOTAL DISCHARGE

Type: SeriesSlot

Units: FLOW

Description: The actual flow at the control point including upstream inflows and local inflows. (Different than Outflow slot)

Information:

I/O: Output only

Links: Not Linkable

9.1.8 Flood Control

Selection of a flood control method in this category makes accessible the appropriate options to have the control point participate in flood control operations as specified in the Flood Control category on the containing computational subbasin. The method names available on the control point for this category correspond to the method names under Flood Control on the computational subbasin so the user can easily select the appropriate control point method to correspond to the type of flood control picked for the containing subbasin.

The following table demonstrates what dependent method categories become available to the user upon selection of Flood Control methods on the control point.

Method	Dependent Method Category	Input Slots	Output Slots
None	(none)	(none)	(none)
Operating Level Balancing		(none)	(none)
	Regulation Discharge	(described below)	(described below)
Phase Balancing		(none)	(none)
	Key Control Point Balancing	(described below)	(described below)

9.1.8.1 None

This method is the default for the Flood Control category and means that no flood control options are selected for the control point. No slots are associated with this method.

9.1.8.2 Operating Level Balancing

Selecting Operating Level Balancing means that options for the control point that are applicable to the Operating Level Balancing method of flood control are made visible. This includes the Regulation Discharge category, and depending on the selection there, associated dependent categories and method options as discussed in subsequent sections. No slots are associated with the Operating Level Balancing method selection. Use of this method for USACE-SWD is described HERE (USACE_SWD.pdf, Section 3.6).

9.1.8.3 Phase Balancing

Selecting Phase Balancing means that options for the control point that are applicable to the Phase Balancing method of flood control are made visible. This includes the Key Control Point Balancing category and its method that is applicable to this type of flood control. Key Control Point Balancing is discussed in a subsequent section. No slots are associated with the Phase Balancing method selection.

9.1.9 Regulation Discharge

The Regulation Discharge method category is only visible if the Operating Level Balancing method is selected for the Flood Control category on the control point. A selected regulation discharge method will calculate regulation discharge, meaning the maximum flow that is allowed in the channel at the control point, for each timestep in the specified forecast period. Calculation of these values are fundamental to the Operating Level Balancing procedure for determining flood control releases.

At the selection of any regulation discharge method, categories for Stage Control Over Forecast, Sag Operation and Regulation Recession appear. Selecting methods in these dependent categories can alter the regulation discharge results as described in detail below.

Selecting a Regulation Discharge method also makes visible categories for Key Control Point Balancing and Flooding Exception. Methods in these categories do not modify regulation discharge values, but regulation discharge values are necessary as inputs to these methods.

Regulation discharge methods are only executed when the regulation discharge (G) flag is set for a timestep on the Reg Discharge Calculation slot. This flag can only be set by a rule. When a rule has set the flag for a timestep and the inflow to the control point is known, the solveOutflowGivenInflow dispatch method calls the selected regulation discharge method for execution. After execution the flag is removed so that regulation discharge will not redispatch unless the flag is reset by a rule. In this way the timing of calculating regulation discharge can be controlled by the order of the ruleset. This ordering (for USACE-SWD models is described HERE (USACE_SWD.pdf, Section 3.5).

The following table summarizes options for regulation discharge and its dependent method categories. It shows dependencies among methods and the slots associated with each. All are described in detail in subsequent sections.

Method	Dependent Method Category	Dependent Method	Input Slots	Output Slots
None	(none)		(none)	(none)
Channel Regulation			Discharge Table Stage Control Intervals Additional Peaking Flow	Reg Discharge Calculation Regulation Discharge Empty Space
	Stage Control Over Forecast	Stage Control Varies Over Forecast	(none)	(none)
		Stage Control Fixed Over Forecast	(none)	(none)
	Sag Operation	None	(none)	(none)
		Sag Operation	Sag Period Sag Tolerance	Sag Operation
	Regulation Recession	None	(none)	(none)

Method	Dependent Method Category	Dependent Method	Input Slots	Output Slots
		Regulation Recession	Regulation Recession	(none)
	Key Control Point Balancing	None	(none)	(none)
		Operating Level Balancing	Key Control Point Reservoirs Balance Period Balance Tolerance Balance Iterations	Balance Level Share
	Flooding Exception	None	(none)	(none)
		Flooding Exception	Excepted Reservoirs Excepted Flood Releases	(none)
	Hydropower Flooding Exception	None	(none)	(none)
		Releases Not Constrained by Flooding	(none)	(none)
Reservoir Current Level Regulation			Regulation Reservoirs Level Regulation Table Stage Control Intervals Variable Regulation Intervals Variable Regulation Tolerance Additional Peaking Flow Level Regulation Parameter Minimum	Reg Discharge Calculation Regulation Discharge Empty Space Level Regulation Parameter
	Stage Control Over Forecast	Stage Control Varies Over Forecast	(none)	(none)
		Stage Control Fixed Over Forecast	(none)	(none)
	Sag Operation	None	(none)	(none)
		Sag Operation	Sag Period Sag Tolerance	Sag Operation
	Regulation Recession	None	(none)	(none)
		Regulation Recession	Regulation Recession	(none)
	Key Control Point Balancing	None	(none)	(none)
		Operating Level Balancing	Key Control Point Reservoirs Balance Period Balance Tolerance Balance Iterations	Balance Level Share
	Flooding Exception	None	(none)	(none)

Method	Dependent Method Category	Dependent Method	Input Slots	Output Slots
	Hydropower Flooding Exception	None	(none)	(none)
		Releases Not Constrained by Flooding	(none)	(none)
		Flooding Exception	Excepted Reservoirs Excepted Flood Releases	(none)
Reservoir Future Level Regulation			Regulation Reservoirs Level Regulation Table Future Level Tolerance Future Level Iterations Stage Control Intervals Variable Regulation Intervals Variable Regulation Tolerance Additional Peaking Flow Level Regulation Parameter Minimum	Reg Discharge Calculation Regulation Discharge Empty Space Level Regulation Parameter
	Stage Control Over Forecast	Stage Control Varies Over Forecast	(none)	(none)
		Stage Control Fixed Over Forecast	(none)	(none)
	Sag Operation	None	(none)	(none)
		Sag Operation	Sag Period Sag Tolerance	Sag Operation
	Regulation Recession	None	(none)	(none)
		Regulation Recession	Regulation Recession	(none)
	Key Control Point Balancing	None	(none)	(none)
		Operating Level Balancing	Key Control Point Reservoirs Balance Period Balance Tolerance Balance Iterations	Balance Level Share
	Flooding Exception	None	(none)	(none)
		Flooding Exception	Excepted Reservoirs Excepted Flood Releases	(none)
	Hydropower Flooding Exception	None	(none)	(none)
		Releases Not Constrained by Flooding	(none)	(none)

Method	Dependent Method Category	Dependent Method	Input Slots	Output Slots
Reservoir System Percent Full Regulation			Regulation Reservoirs Percent Full Regulation Table Minimum Regulation Discharge Stage Control Intervals Variable Regulation Intervals Variable Regulation Tolerance Additional Peaking Flow	Reg Discharge Calculation Regulation Discharge Empty Space Hydrograph
	Percent Full Determination	Flood and Surcharge Water	(none)	(none)
		Flood Water Only	(none)	(none)
	Stage Control Over Forecast	Stage Control Varies Over Forecast	(none)	(none)
		Stage Control Fixed Over Forecast	(none)	(none)
	Sag Operation	None	(none)	(none)
		Sag Operation	Sag Period Sag Tolerance	Sag Operation
	Regulation Recession	None	(none)	(none)
		Regulation Recession	Regulation Recession	(none)
	Key Control Point Balancing	None	(none)	(none)
		Operating Level Balancing	Key Control Point Reservoirs Balance Period Balance Tolerance Balance Iterations	Balance Level Share
	Flooding Exception	None	(none)	(none)
		Flooding Exception	Excepted Reservoirs Excepted Flood Releases	(none)
	Hydropower Flooding Exception	None	(none)	(none)
		Releases Not Constrained by Flooding	(none)	(none)

9.1.9.1 None

This default method means no regulation discharge information is calculated and the control point cannot be part of operating level balancing for flood control.

9.1.9.2 Channel Regulation

This method calculates regulation discharge based on information associated with the channel and not on conditions in upstream reservoirs. Regulation discharge calculations depend on discharge information entered by the user and can optionally be modified over certain periods by specifying intervals of stage control. The method populates results for the forecast period into the Regulation Discharge and Empty Space series slots.

SLOTS SPECIFIC TO THIS METHOD

☞ **DISCHARGE TABLE**

Type: PeriodicSlot

Units: FLOW

Description: Schedule of discharge values the user wants to use for determining regulation discharge with different rows representing different time points in the schedule, and different columns containing different discharge values.

Information: This is a n x n periodic slot where the user specifies the period over which the schedule repeats (daily, yearly, etc.) and the slot has the capability to generate discharges for any date from the schedule.

I/O: Required input

Links: Not linkable

☞ **STAGE CONTROL INTERVALS**

Type: Periodic Slot

Units: FLOW

Description: User-defined intervals of stage control that are specified by pairing dates with lower discharge bound values.

Information: An interval of stage control runs from one specified date to the following specified date, and the lower discharge bound is by default a lookup to the value specified with the first date. If there is one interval only, or nonadjacent intervals, a NAN value can be paired with the date in the following line to signify the end of the stage control interval.

I/O: Optional Input

Links: Not linkable

☞ Reg Discharge Calculation

Type: SeriesSlot

Units: NONE

Description: This slot controls and documents the calculation of regulation discharge.

Information: The G flag must be set by rule for a timestep on this slot before regulation discharge is called by the dispatch method. The G flag gets removed when the regulation discharge calculation is complete. The value on the slot records the number of times that the G flag was set and regulation discharge was calculated for a timestep.

I/O: Output only

Links: Not linkable

☞ Regulation Discharge

Type: SeriesSlot

Units: FLOW

Description: Regulation discharge values that are calculated at each time step.

Information: Contains results for each simulation timestep and also acts to temporarily hold the regulation results for the associated forecast timesteps during simulation of the current simulation timestep.

I/O: Output only

Links: Not linkable

☞ Empty Space

Type: SeriesSlot

Units: FLOW

Description: Amount of space calculated as available for additional flow in the channel.

Information: Contains results for each simulation timestep and also acts to temporarily hold the empty space data for the associated forecast timesteps during simulation of the current simulation timestep.

I/O: Output only

Links: Not linkable

☞ Additional Peaking Flow

Type: SeriesSlot

Units: FLOW

Description: Difference between the estimated peak flow through the control point (average over the timestep) and the instantaneous peak during the timestep.

Information: Additional Peaking Flow is used in the Empty Space calculation. If users have input a value for Additional Peaking Flow, the Empty Space will be reduced by this additional amount. Typically, the slot contains zeros for most timesteps and peak values on certain timesteps with peaks.

I/O: Optional; if not input, Additional Peaking Flow is not considered

Links: Usually not linked

The Channel Regulation Discharge method executes the following for all timesteps in the forecast period:

Check the Stage Control Intervals slot to determine if any forecast timesteps fall in an interval of Stage Control. If any or all do, execute the following Stage Control Calculations to assign regulation discharge values for those timesteps.

- Construct a list of discharge values using the lower bound defined for the interval on the Stage Control Intervals slot, and all discharge values greater than this lower bound that are defined for this date in the Discharge Table slot.
- If the previous simulation timestep's inflow at the control point is less than the lower bound for the interval, the regulation discharge is set to the lower bound value.
- If not, the regulation discharge is set to the largest discharge in the list that is equaled or exceeded by the larger of the previous simulation timestep's regulation discharge and the previous simulation timestep's inflow at the control point.

For any timesteps that are not in Stage Control, set the regulation discharge to the minimum value in the Discharge Table for the date.

Execute default method for Sag Operation method category.

Execute default method for Regulation Recession method category.

Calculate Empty Space for each forecast timestep as the Regulation Discharge minus the flow through the control point. Flow through the control point is the sum of Inflow, Local Inflow and Additional Peaking Flow.

9.1.9.3 Reservoir Current Level Regulation

This method calculates regulation discharge based on the current (previous timestep) level of a single associated upstream reservoir and a Level Regulation Table that associates the level with discharge values by date. Regulation discharge calculations can optionally be modified over certain periods by specifying intervals of stage control or intervals of variable regulation. The method populates results for the forecast period into the Regulation Discharge and Empty Space series slots.

Slots Specific to this Method

☞ Stage Control Intervals

Type: Periodic Slot

Units: FLOW

Description: User-defined intervals of stage control that are specified by pairing dates with lower discharge bound values.

Information: An interval of stage control runs from one specified date to the following specified date, and the lower discharge bound is by default a lookup to the value specified with the first date. If there is one interval only, or nonadjacent intervals, a NAN value can be paired with the date in the following line to signify the end of the stage control interval.

I/O: Optional input

Links: Not linkable

☞ Reg Discharge Calculation

Type: SeriesSlot

Units: NONE

Description: This slot controls and documents the calculation of regulation discharge.

Information: The G flag must be set by rule for a timestep on this slot before regulation discharge is called by the dispatch method. The G flag gets removed when the regulation discharge calculation is complete. The value on the slot records the number of times that the G flag was set and regulation discharge was calculated for a timestep.

I/O: Output only

Links: Not linkable

☞ Regulation Discharge

Type: SeriesSlot

Units: FLOW

Description: Regulation discharge values that are calculated at each time step.

Information: Contains results for each simulation timestep and also acts to temporarily hold the regulation results for the associated forecast timesteps during simulation of the current simulation timestep.

I/O: Output only

Links: Not Linkable

☞ Empty Space

Type: SeriesSlot

Units: FLOW

Description: Amount of space calculated as available for additional flow in the channel.

Information: Contains results for each simulation timestep and also acts to temporarily hold the empty space data for the associated forecast timesteps during simulation of the current simulation timestep.

I/O: Output only

Links: Not linkable

☞ Additional Peaking Flow

Type: SeriesSlot

Units: FLOW

Description: Difference between the estimated peak flow through the control point (average over the timestep) and the instantaneous peak during the timestep.

Information: Additional Peaking Flow is used in the Empty Space calculation. If users have input a value for Additional Peaking Flow, the Empty Space will be reduced by this additional amount. Typically, the slot contains zeros for most timesteps and peak values on certain timesteps with peaks.

I/O: Optional; if not input, Additional Peaking Flow is not considered

Links: Usually not linked

☞ Regulation Reservoirs

Type: ListSlot

Units: NONE

Description: Name of the single associated upstream reservoir used in the regulation discharge calculations.

Information:

I/O: Required input

Links: Not linkable

☞ Variable Regulation Intervals

Type: PeriodicSlot

Units: FLOW

Description: User-defined intervals of variable regulation that are specified by dates with a lower discharge bound, an upper discharge bound, and a maximum discharge.

Information: An interval of variable regulation runs from one specified date to the following specified date, with the lower and upper discharge bounds and maximum discharge by default being a lookup to the values specified with the first date. If there is one interval only, or nonadjacent intervals, NAN values can be specified with the date in the following line to signify the end of the variable regulation interval.

I/O: Optional input

Links: Not linkable

☞ Variable regulation Tolerance

Type: ScalarSlot

Units: DECIMAL

Description: Tolerance (as a proportion) on comparison of the previous simulation timestep's total flow with previous simulation timestep's regulation discharge; if difference is greater than this tolerance, the current regulation discharge is reset to the previous simulation timestep's total flow.

Information: Default value of 0.03 comes from the 3% value coded into COE's SUPER program.

I/O: Optional input

Links: Not linkable

☞ Level Regulation Table

Type: PeriodicSlot

Units: NONE

Description: Schedule of reservoir level values where rows represent times, columns represent discharges and cell values represent the level value of the associated reservoir.

Information: This is an n x n periodic slot where the user specifies the period over which the schedule repeats (daily, yearly, etc.) and the slot has the capability to do a reverse lookup to generate a discharge value given a date and a reservoir level.

I/O: Required input

Links: Not linkable

☞ Level Regulation Parameter Minimum

Type: Periodic

Units: None

Description: This is the lowest possible allowed Level Regulation Parameter

Information: If no values are specified in this slot, the lowest possible Level Regulation Parameter will be the Top of Conservation Pool on the specified reservoir.

I/O: Optional input

Links: Not linkable

☞ Level Regulation Parameter

Type: Series Slot

Units: NONE

Description: This is the operating level that is used in the Level Regulation Table to determine the discharge.

Information: Only the current timesteps value is set

I/O: Output only

Links: Not Linkable

> **Note:** If the reservoir is disabled and just passing inflows (**HERE (Objects.pdf, Section 24.1.30.2)**), the Regulation Discharge is set to the highest specified discharge in the Level Regulation Table. This is the largest flow in the column headings. This large Regulation Discharge is set so that this control point does not limit any other upstream reservoirs.

The Reservoir Current Level Regulation Discharge method executes the following for all timesteps in the forecast period:

Check the Stage Control Intervals slot to determine if any forecast timesteps fall in an interval of Stage Control. If any or all do, execute the following Stage Control Calculations to assign regulation discharge values for those timesteps.

- Construct a list of discharge values using the lower bound defined for the interval on the Stage Control Intervals slot, and all discharge values greater than this lower bound that are defined in the Level Regulation Table slot.
- If the previous simulation timestep's inflow at the control point is less than the lower bound for the interval, the regulation discharge is set to the lower bound value.
- If not, the regulation discharge is set to the largest discharge in the list that is equaled or exceeded by the larger of the previous simulation timestep's regulation discharge and the previous simulation timestep's inflow at the control point.

For any timesteps that are not in Stage Control, the Level Regulation Parameter is set as the previous simulation timestep's reservoir level. This value is then looked up in the Level Regulation Table using the forecast timestep's date to generate a regulation discharge.

Note: If the computed Level Regulation Parameter is less than the Level Regulation Parameter Minimum for the specified date, it is reset to the Level Regulation Parameter Minimum. If the Level Regulation Parameter Minimum periodic slot is not populated, the reservoir's top of the conservation pool level is used as the minimum.

For any timesteps that are not in Stage Control, check the Variable Regulation Intervals slot to determine if these timesteps fall in an interval of Variable Regulation. If the current forecast timestep is within the start and end times of a variable regulation interval and the current regulation discharge falls between the lower and upper bounds defined on the Variable Regulation Intervals slot, do the following:

- Reset the regulation discharge to the lower bound value.
- If the previous forecast timestep's regulation discharge is larger than the current regulation discharge, the current regulation discharge is reset to the previous timestep's value.
- If the previous simulation timestep's total inflow is 1 plus the value in the Variable Regulation Tolerance Slot (default is 0.03) times larger than the previous forecast timestep's regulation discharge, then the current regulation discharge is reset to the previous simulation timestep's total inflow.
- Finally, the maximum discharge check is applied. If the current regulation discharge exceeds the maximum discharge specified in the Variable Regulation Intervals slot, the regulation discharge is reset to the maximum discharge value.

Execute default method for Sag Operation method category.

Execute default method for Regulation Recession method category.

Calculate Empty Space for each forecast timestep as the Regulation Discharge minus the flow through the control point. Flow through the control point is the sum of Inflow, Local Inflow and Additional Peaking Flow.

9.1.9.4 Reservoir Future Level Regulation

This method calculates regulation discharge based on the future level over the forecast period of a single associated upstream reservoir and a Level Regulation Table that associates the level with discharge values by date. Regulation discharge calculations can optionally be modified over certain periods by specifying intervals of stage control or intervals of variable regulation. The method populates results for the forecast period into the Regulation Discharge and Empty Space series slots.

SLOTS SPECIFIC TO THIS METHOD

Stage Control Intervals

Type: Periodic Slot

Units: FLOW

Description: User-defined intervals of stage control that are specified by pairing dates with lower discharge bound values.

Information: An interval of stage control runs from one specified date to the following specified date, and the lower discharge bound is by default a lookup to the value specified with the first date. If there is one interval only, or nonadjacent intervals, a NAN value can be paired with the date in the following line to signify the end of the stage control interval.

I/O: Optional input

Links: Not linkable

Reg Discharge Calculation

Type: SeriesSlot

Units: NONE

Description: This slot controls and documents the calculation of regulation discharge.

Information: The G flag must be set by rule for a timestep on this slot before regulation discharge is called by the dispatch method. The G flag gets removed when the regulation discharge calculation is complete. The value on the slot records the number of times that the G flag was set and regulation discharge was calculated for a timestep.

I/O: Output only

Links: Not linkable

Regulation Discharge

Type: SeriesSlot

Units: FLOW

Description: Regulation discharge values that are calculated at each time step.

Information: Contains results for each simulation timestep and also acts to temporarily hold the regulation results for the associated forecast timesteps during simulation of the current simulation timestep.

I/O: Output only

Links: Not Linkable

Empty Space

Type: SeriesSlot

Units: FLOW

Description: Amount of space calculated as available for additional flow in the channel.

Information: Contains results for each simulation timestep and also acts to temporarily hold the empty space data for the associated forecast timesteps during simulation of the current simulation timestep.

I/O: Output only

Links: Not linkable

☞ Additional Peaking Flow

Type: SeriesSlot

Units: FLOW

Description: Difference between the estimated peak flow through the control point (average over the timestep) and the instantaneous peak during the timestep.

Information: Additional Peaking Flow is used in the empty space calculation. If users have input a value for Additional Peaking Flow, the empty space will be reduced by this additional amount. Typically, the slot contains zeros for most timesteps and peak values on certain timesteps with peaks.

I/O: Optional; if not input, Additional Peaking Flow is not considered

Links: Usually not linked

☞ Regulation Reservoirs

Type: ListSlot

Units: NONE

Description: Name of the single associated upstream reservoir used in the regulation discharge calculations.

Information:

I/O: Required input

Links: Not linkable

☞ Variable Regulation Intervals

Type: PeriodicSlot

Units: FLOW

Description: User-defined intervals of variable regulation that are specified by dates with a lower discharge bound, an upper discharge bound, and a maximum discharge.

Information: An interval of variable regulation runs from one specified date to the following specified date, with the lower and upper discharge bounds and maximum discharge by default being a lookup to the values specified with the first date. If there is one interval only, or nonadjacent intervals, NAN values can be specified with the date in the following line to signify the end of the variable regulation interval.

I/O: Optional input

Links: Not linkable

☞ VARIABLE REGULATION TOLERANCE

Type: ScalarSlot

Units: DECIMAL

Description: Tolerance (as a proportion) on comparison of the previous simulation timestep's total flow with the previous simulation timestep's regulation discharge; if difference is greater than this tolerance, the current regulation discharge is reset to the previous simulation timestep's total flow.

Information: Default value of 0.03 comes from the 3% value coded into COE's SUPER program.

I/O: Optional input

Links: Not linkable

☞ LEVEL REGULATION TABLE

Type: PeriodicSlot

Units: NONE

Description: Schedule of reservoir level values where rows represent times, columns represent discharges and cell values represent the level value of the associated reservoir.

Information: This is an n x n periodic slot where the user specifies the period over which the schedule repeats (daily, yearly, etc.) and the slot has the capability to do a lookup to generate a reservoir level given a date and a discharge value.

I/O: Required input

Links: Not linkable

☞ LEVEL REGULATION PARAMETER MINIMUM

Type: Periodic

Units: NONE

Description: This is the lowest possible allowed Level Regulation Parameter

Information: If no values are specified in this slot, the lowest possible Level Regulation Parameter will be the Top of Conservation Pool on the specified reservoir.

I/O: Optional input

Links: Not linkable

☞ Level Regulation Parameter

Type: Series Slot

Units: None

Description: This is the operating level that is used in the Level Regulation Table to determine the discharge.

Information: A value is only set for the current timestep

I/O: Output only

Links: Not Linkable

☞ Future Level Tolerance

Type: ScalarSlot

Units: NONE

Description: Tolerance for the user to specify how close a future predicted maximum level for a reservoir assuming a certain discharge needs to be compared to the level for that discharge from the level regulation table. The level tolerance is expressed as a difference between the levels.

Information: The algorithm to find a future predicted maximum level for a reservoir that equals the level for a discharge from the level regulation table will iterate until the tolerance is reached or until the maximum number of iterations is reached, whichever comes first. Serves to limit excessive iterations in the calculation algorithm for reservoir future level regulation.

I/O: Required input

Links: Not linkable

☞ Future Level Iterations

Type: ScalarSlot

Units: NONE

Description: Maximum number of iterations for the reservoir future level regulation algorithm.

Information: The algorithm to find a future predicted maximum level for a reservoir that equals the level for a discharge from the level regulation table will iterate until the tolerance specified in the Future Level Tolerance slot is reached or until the maximum number of iterations is reached, whichever comes first.

Serves to limit excessive iterations in the calculation algorithm for reservoir future level regulation.

I/O: Required input

Links: Not linkable

Note: If the reservoir is disabled and just passing inflows (**HERE (Objects.pdf, Section 24.1.30.2)**), the Regulation Discharge is set to the highest specified discharge in the Level Regulation Table. This is the largest flow in the column headings. This large Regulation Discharge is set so that this control point does not limit any other upstream reservoirs.

The Reservoir Future Level Regulation Discharge method executes the following for all timesteps in the forecast period:

Check the Stage Control Intervals slot to determine if any forecast timesteps fall in an interval of Stage Control. If any or all do, execute the following Stage Control Calculations to assign regulation discharge values for those timesteps.

- Construct a list of discharge values using the lower bound defined for the interval on the Stage Control Intervals slot, and all discharge values greater than this lower bound that are defined in the Level Regulation Table slot.
- If the previous simulation timestep's inflow at the control point is less than the lower bound for the interval, the regulation discharge is set to the lower bound value.
- If not, the regulation discharge is set to the largest discharge in the list that is equaled or exceeded by the larger of the previous simulation timestep's regulation discharge and the previous simulation timestep's inflow at the control point.

If any forecast timesteps are not in stage control, calculate the Level Regulation Parameter. This involves finding future maximum levels in the one associated reservoir at various discharges and finding the one that matches the level for the current simulation timestep at that discharge in the Regulation Level Schedule table according to the following steps:

- Get the level from the Level Regulation Table corresponding to the lowest discharge value and the current simulation date. Calculate the highest level the reservoir would reach over the forecast period assuming this discharge on every timestep and including all forecasted inflows, but not including surcharge releases. Compare this level to the level from the schedule table. If it is less than the table value, the level parameter for the solution is the calculated forecasted level. If it is greater than the table level, record this lowest discharge value and move to the next step.
- Take the highest discharge value from the Level Regulation Table and repeat the procedure in the above step. If the highest forecasted level is greater than the level from the table for this highest discharge value, the level parameter for the solution is set to the highest predefined operating level for the reservoir. If the highest forecasted level is less, record the highest discharge value and move to the next step.

- Perform a bisection search algorithm using the lowest and highest discharge values as initial end points. Do the forecasted level and table level calculations as above for the midpoint discharge value and reset it to the appropriate end point based on the results. Iterate this algorithm until the forecasted level and table level are within the specified tolerance in the Future Level Tolerance slot, or until the maximum number of specified iterations is reached from the Future Level Iterations slot.
- If the maximum number of iterations is reached, a warning is issued to the user to let them know that iterations were exceeded without reaching the desired tolerance. Processing proceeds in this case using the level from the last iteration.
- In all cases, the Level Regulation Parameter for the solution is limited to the highest predefined operating level for the reservoir.
- Also, the Level Regulation Parameter is limited to be greater than the Level Regulation Parameter Minimum for the specified date. If the Level Regulation Parameter Minimum periodic slot is not populated, the reservoir's top of the conservation pool level is used as the minimum.
- For each timestep in the forecast period that is not in stage control, the regulation discharge is determined by looking up the Level Regulation Parameter and the current forecast timestep in the Level Regulation Table.

For any timesteps that are not in Stage Control, check the Variable Regulation Intervals slot to determine if these timesteps fall in an interval of Variable Regulation. If the current forecast timestep is within the start and end times of a variable regulation interval and the current regulation discharge falls between the lower and upper bounds defined on the Variable Regulation Intervals slot, do the following:

- Reset the regulation discharge to the lower bound value.
- If the previous forecast timestep's regulation discharge is larger than the current regulation discharge, the current regulation discharge is reset to the previous timestep's value.
- If the previous simulation timestep's total inflow is 1 plus the value in Variable Regulation Tolerance Slot (default is 0.03) times larger than the previous forecast timestep's regulation discharge, then the current regulation discharge is reset to the previous simulation timestep's total inflow.
- Finally, the maximum discharge check is applied. If the current regulation discharge exceeds the maximum discharge specified in the Variable Regulation Intervals slot, the regulation discharge is reset to the maximum discharge value.

Execute default method for Sag Operation method category.

Execute default method for Regulation Recession method category.

Calculate Empty Space for each forecast timestep as the Regulation Discharge minus the flow through the control point. Flow through the control point is the sum of Inflow, Local Inflow and Additional Peaking Flow.

9.1.9.5 Reservoir System Percent Full Regulation

This method calculates regulation discharge based on the percent of the flood pool occupied over the forecast period in a system of one or more associated reservoirs and a Percent Full Regulation Table that associates the percent full values with discharge values by date. Regulation discharge calculations can optionally be modified over certain periods by specifying intervals of stage control or intervals of variable regulation. The method populates results for the forecast period into the Regulation Discharge and Empty Space series slots.

SLOTS SPECIFIC TO THIS METHOD

☞ STAGE CONTROL INTERVALS

Type: Periodic Slot

Units: FLOW

Description: User-defined intervals of stage control that are specified by pairing dates with lower discharge bound values.

Information: An interval of stage control runs from one specified date to the following specified date, and the lower discharge bound is by default a lookup to the value specified with the first date. If there is one interval only, or nonadjacent intervals, a NAN value can be paired with the date in the following line to signify the end of the stage control interval.

I/O: Optional input

Links: Not linkable

☞ REG DISCHARGE CALCULATION

Type: SeriesSlot

Units: NONE

Description: This slot controls and documents the calculation of regulation discharge.

Information: The G flag must be set by rule for a timestep on this slot before regulation discharge is called by the dispatch method. The G flag gets removed when the regulation discharge calculation is complete. The value on the slot records the number of times that the G flag was set and regulation discharge was calculated for a timestep.

I/O: Output only

Links: Not linkable

☞ Regulation Discharge

Type: SeriesSlot

Units: FLOW

Description: Regulation discharge values that are calculated at each time step.

Information: Contains results for each simulation timestep and also acts to temporarily hold the regulation results for the associated forecast timesteps during simulation of the current simulation timestep.

I/O: Output only

Links: Not Linkable

☞ Empty Space

Type: SeriesSlot

Units: FLOW

Description: Amount of space calculated as available for additional flow in the channel.

Information: Contains results for each simulation timestep and also acts to temporarily hold the empty space data for the associated forecast timesteps during simulation of the current simulation timestep.

I/O: Output only

Links: Not linkable

☞ Additional Peaking Flow

Type: SeriesSlot

Units: FLOW

Description: Difference between the estimated peak flow through the control point (average over the timestep) and the instantaneous peak during the timestep.

Information: Additional Peaking Flow is used in the empty space calculation. If users have input a value for Additional Peaking Flow, the empty space will be reduced by this additional amount. Typically, the slot contains zeros for most timesteps and peak values on certain timesteps with peaks.

I/O: Optional; if not input, Additional Peaking Flow is not considered

Links: Usually not linked

☞ Regulation Reservoirs

Type: ListSlot

Units: NONE

Description: Name of the associated upstream reservoirs used in the regulation discharge calculations.

Information:

I/O: Required input

Links: Not linkable

☞ Variable Regulation Intervals

Type: PeriodicSlot

Units: FLOW

Description: User-defined intervals of variable regulation that are specified by dates with a lower discharge bound, an upper discharge bound, and a maximum discharge.

Information: An interval of variable regulation runs from one specified date to the following specified date, with the lower and upper discharge bounds and maximum discharge by default being a lookup to the values specified with the first date. If there is one interval only, or nonadjacent intervals, NAN values can be specified with the date in the following line to signify the end of the variable regulation interval.

I/O: Optional input

Links: Not linkable

☞ Variable regulation Tolerance

Type: ScalarSlot

Units: DECIMAL

Description: Tolerance (as a proportion) on comparison of the previous simulation timestep's total flow with the previous simulation timestep's regulation discharge; if difference is greater than this tolerance, the current regulation discharge is reset to the previous simulation timestep's total flow.

Information: Default value of 0.03 comes from the 3% value coded into COE's SUPER program.

I/O: Optional input

Links: Not linkable

☞ Percent Full Regulation Table

Type: PeriodicSlot

Units: DECIMAL

Description: Schedule of percent full values where rows represent times, columns represent discharges and cell values represent the percent full value (as a decimal) of the associated reservoirs.

Information: This is a n x n periodic slot where the user specifies the period over which the schedule repeats (daily, yearly, etc.) and the slot has the capability to do a reverse lookup to generate a discharge value given a date and a percent full value.

I/O: Required input

Links: Not linkable

☞ Percent Full Regulation Parameter

Type: Series Slot

Units: Decimal

Description: This is the percentage of full used in the Percent Full Regulation Table.

Information: A value is set for each timestep in the forecast period

I/O: Output only

Links: Not linkable

☞ Minimum Regulation Discharge

Type: ScalarSlot

Units: FLOW

Description: Minimum regulation discharge value used in calculating the percent full parameter.

Information: This is a single value in a 1x1 table slot and is used only in the Reservoir System Percent Full Regulation method

I/O: Optional input

Links: Not Linkable

Note: If the reservoir is disabled and just passing inflows (**HERE (Objects.pdf, Section 24.1.30.2)**), the reservoir is skipped in the computations below. The flows into that reservoir should already be passing through the reservoir.

The Reservoir System Percent Full Regulation Discharge method executes the following for all timesteps in the forecast period:

Check the Stage Control Intervals slot to determine if any forecast timesteps fall in an interval of Stage Control. If any or all do, execute the following Stage Control Calculations to assign regulation discharge values for those timesteps.

- Construct a list of discharge values using the lower bound defined for the interval on the Stage Control Intervals slot, and all discharge values greater than this lower bound that are defined in the Percent Full Regulation Table slot.
- If the previous simulation timestep's inflow at the control point is less than the lower bound for the interval, the regulation discharge is set to the lower bound value.
- If not, the regulation discharge is set to the largest discharge in the list that is equaled or exceeded by the larger of the previous simulation timestep's regulation discharge and the previous simulation timestep's inflow at the control point.

If any forecast timesteps are not in stage control, calculate the Percent Full Regulation Parameter as follows:

- The total flood pool capacity of the system is determined as the difference between storage at top of flood pool and storage at conservation pool at the current simulation timestep for all the reservoirs listed in the Regulation Reservoirs slot. This is used as the denominator in all the following percent full calculations.
- In calculating the quantity of water in flood pool storage for the various timesteps in the following calculations, the storage at conservation pool for the current simulation timestep is used as the baseline storage value for determining flood pool volumes.
- When this regulation discharge method is selected, a dependent method category appears called Percent Full Determination. The selected method in this category determines if flood and surcharge water are included in flood pool storage calculations or if flood water only is included. The default method is flood and surcharge water to match the way COE's SUPER program performs this calculation.
- Percent Full Regulation Parameters values for the current timestep and for the remaining forecast timesteps are calculated as follows. The numerator is the total flood pool storage on the desired timestep for all reservoirs in the Regulation Reservoirs list modified by potential adjustments as follows. For each timestep in the forecast period up to the desired timestep, if the total inflow to the control point for a timestep is less than the value specified in the optional Minimum Regulation Discharge slot, the inflow is subtracted from this minimum value and the numerator is reduced by this difference. After this adjustment is made for each timestep up to the desired timestep, a percent full value for the desired timestep is then calculated by dividing the adjusted numerator by the denominator determined above.
- The maximum percent full value for the system is then determined from among the ones calculated for each timestep in the forecast.
- For each timestep in the forecast period that is not in stage control, the regulation discharge is determined by looking up the maximum percent full value and the current forecast timestep in the Percent Full Regulation Table. If the maximum percent full

value exceeds the highest percent defined for that date in the table, the regulation discharge is assigned to be the highest discharge value defined in the table.

For any timesteps that are not in Stage Control, check the Variable Regulation Intervals slot to determine if these timesteps fall in an interval of Variable Regulation. If the current forecast timestep is within the start and end times of a variable regulation interval and the current regulation discharge falls between the lower and upper bounds defined on the Variable Regulation Intervals slot, do the following:

- Reset the regulation discharge to the lower bound value.
- If the previous forecast timestep's regulation discharge is larger than the current regulation discharge, the current regulation discharge is reset to the previous timestep's value.
- If the previous simulation timestep's total inflow is 1 plus the value in the Variable Regulation Tolerance Slot (default is 0.03) times larger than the previous forecast timestep's regulation discharge, then the current regulation discharge is reset to the previous simulation timestep's total inflow.
- Finally, the maximum discharge check is applied. If the current regulation discharge exceeds the maximum discharge specified in the Variable Regulation Intervals slot, the regulation discharge is reset to the maximum discharge value.

Execute default method for Sag Operation method category.

Execute default method for Regulation Recession method category.

Calculate Empty Space for each forecast timestep as the Regulation Discharge minus the flow through the control point. Flow through the control point is the sum of Inflow, Local Inflow and Additional Peaking Flow.

9.1.10 Percent Full Determination

This category appears only when the Reservoir System Percent Full Regulation method is selected in the Regulation Discharge category. The selection here determines what type of water is included in the numerator for the percent full calculations performed in that regulation discharge method.

9.1.10.1 Flood and Surcharge Water

Selecting this method means that flood pool and surcharge pool water will be included in the numerator of the reservoir system percent full calculations. Since the denominator in these calculations is only the capacity of the flood pools of all the reservoirs in the system, including surcharge water in the numerator means that the percent full for the system can be greater than 100%. This is the default method selection because this is the way that COE's SUPER program performs this calculation. There are no slots specific to this method.

9.1.10.2 Flood Water Only

Selecting this method means that flood pool water only is included in the numerator of the reservoir system percent full calculations. Since the denominator in these calculations is the flood pool capacity of all the reservoirs in the system, the maximum possible percent full is 100%. There are no slots specific to this method.

9.1.11 Stage Control Over Forecast

This category appears at the selection of any regulation discharge method in the Regulation Discharge category, so is related to the operating level balancing approach to flood control. The selection here determines how the stage control part of the regulation discharge calculation is applied over the forecast period.

9.1.11.1 Stage Control Varies Over Forecast

This default method selection means that each forecast timestep during the execution of regulation discharge is compared to the Stage Control Intervals periodic slot under the regulation discharge method to check if the timestep falls into a specified time interval of stage control. If it does, regulation discharge for that forecast timestep is determined under the stage control portion of the regulation discharge algorithm. If it does not, regulation discharge is calculated without stage control being applied. This determination is made separately for each timestep in the forecast period. There are no slots specific to this method.

9.1.11.2 Stage Control Fixed Over Forecast

This selection means that the current simulation timestep is checked against the Stage Control Intervals periodic slot under the regulation discharge method to determine if the timestep falls into a specified time interval of stage control. If it does, all timesteps in the forecast period are considered to be under stage control. If it does not, none of the timesteps in the forecast period are considered to be under stage control. This is the way that COE’s SUPER program handles this calculation. There are no slots specific to this method.

9.1.12 Sag Operation

This category appears at the selection of any regulation discharge method in the Regulation Discharge category, so is related to the operating level balancing approach to flood control. Once a flood hydrograph recedes to the “normal regulating discharge,” a “sag operation” can be used to reduce the regulating discharge for a few timesteps to allow time for local field drainage. A sag operation will continue for a prescribed number of timesteps during which the regulation discharge will be reduced to the sag discharge. After the sag operation has continued for the specified number of timesteps, operations resume at the normal regulating discharge.

9.1.12.1 None

This default method means no sag operations are performed.

9.1.12.2 Sag Operation

This method implements sag operations. A sag operation is explicitly turned on and continues until it is explicitly turned off, although the regulation discharge is only reset to the reduced sag value for a first number of timesteps equal to the sag period. Sag operations are tracked in the Sag Operation series slot where an incremented number appears in each timestep that is part of a sag operation. Timesteps not in sag operation are assigned zero.

SLOTS SPECIFIC TO THIS METHOD

☞ **SAG PERIOD**

Type: TableSlot

Units: NONE, FLOW

Description: Number of timesteps that the hydrograph should sag below regulation discharge to allow for local field drainage, and the temporary discharge to be used for the sag period.

Information:

I/O: Required input

Links: Not linkable

☞ **SAG OPERATION**

Type: SeriesSlot

Units: NONE

Description: Contains an incremented number for each timestep that is part of a sag operation, allowing tracking of sag operations across simulation timesteps. Timesteps not in sag operation contain a zero.

Information: Other than an initial value to indicate the status of sag operations at the beginning of the run, this slot is output only.

I/O: Output with initial input

Links: Not linkable

☞ Sag Tolerance

Type: ScalarSlot

Units: DECIMAL

Description: When a Sag Operation is ongoing, this is the proportional amount by which the total inflow for the current timestep must be greater than the calculated regulation discharge for current forecast timestep in order for the Sag Operations to be turned off.

Information: The default value of 0.03 is from the 3% value coded into the COE's Super program code.

I/O: Required input

Links: Not linkable

The Sag Operation method does the following:

- Check for an ongoing sag operation. There is an ongoing sag operation if the previous simulation timestep had a nonzero number written to the Sag Operation slot.
- If there is an ongoing sag operation, test to see if it needs to be turned off. If the total inflow for the current simulation timestep is 1 plus the value in the Sag Tolerance Slot (default is 0.03) times greater than the calculated regulation discharge for the current simulation timestep, turn off sag operations and assign zeros to the Sag Operation slot for all timesteps in the forecast period. Regulation discharges for the forecast period will remain unchanged due to sag operations.
- If sag operations are ongoing and are not turned off, continue incrementing the numbers in the Sag Operations slot for all timesteps in the forecast period. Any timesteps in the forecast period that have an incremented number less than or equal to the sag period defined in the Sag Period slot should have their regulation discharges set to the temporary sag discharge value also entered in the Sag Period slot.
- If there is no ongoing sag operation, check to see if one needs to be turned on. A sag period is turned on if the forecast inflow for every timestep in the forecast period is less than that timestep's regulation discharge, and if the previous simulation timestep's total flow is greater than the current simulation timestep's regulation discharge.
- If a sag operation is turned on, assign the number 1 to the Sag Operation slot for the current simulation timestep and assign incremented numbers to all other timesteps in the

forecast period. Any timesteps in the forecast period that have an incremented number less than or equal to the sag period defined in the Sag Period slot should have their regulation discharges set to the temporary sag discharge value also entered in the Sag Period slot.

9.1.13 Regulation Recession

This category appears at the selection of any regulation discharge method in the Regulation Discharge category, so is related to the operating level balancing approach to flood control. It allows the user to limit by how much the regulation discharge recesses from one timestep to the next.

9.1.13.1 None

This default method for the category means no regulation recession is applied.

9.1.13.2 Regulation Recession

This method limits the recession in regulation discharge to the amount specified in the Regulation Recession slot.

Slots Specific to this Method

☞ **Regulation Recession**

Type: PeriodicSlot

Units: FLOW

Description: Schedule of the maximum amount by which the regulation discharge can be receded for a timestep, with different rows representing recessions at different time points in the schedule, and different columns representing recessions at different discharges. Legitimate recession values can be limited to only a certain range of discharges.

Information: This is a n x n periodic slot where the user specifies the period over which the schedule repeats (daily, yearly, etc.) and the table has the capability to generate a recession value for any date and discharge value from the schedule.

I/O: Required input

Links: Not linkable

The Regulation Recession method steps through the timesteps in the forecast period and does the following:

- The current forecast timestep and the current regulation discharge value are looked up in the Regulation Recession table to see if a legitimate regulation recession value applies. If not, regulation discharge for the current timestep is unchanged and subsequent steps are skipped.
- The previous forecast timestep’s regulation discharge is compared to the upper bound of the discharge range where recession applies (on the first forecast timestep, use the

previous simulation timestep's regulation discharge). If it was greater than the upper bound of the range, the current timestep's regulation discharge is reset to this upper bound value.

- If the previous forecast timestep's regulation discharge was within the discharge range where recession applies, the regulation recession value is subtracted from the previous forecast timestep's regulation discharge to get a recessed discharge.
- If the current regulation discharge is more than the recessed discharge, it is unchanged.
- If the current regulation discharge is less than the recessed discharge, the current regulation discharge is reset to the recessed discharge value.

9.1.14 Key Control Point Balancing

This method category can be applicable to both operating level balancing and phase balancing approaches for determining flood control releases. The category appears at the selection of the Phase Balancing method under the Flood Control category of the control point, or at the selection of any regulation discharge method under the Regulation Discharge category that appears under the Operating Level Balancing method for the Flood Control category on the control point.

Selection of a method in this category makes a control point a key control point and causes the key icon to appear on the control point in the GUI to identify it as key. Key control point balancing allows for the balancing of storage in associated upstream reservoirs with respect to the control point. Calculations for the key control point balancing methods are actually initiated from a flood control method selected on a computational subbasin that includes the control point.

9.1.14.1 Operating Level Balancing

This method is only available if one of the regulation discharge methods is selected under the Regulation Discharge category that appears under the Operating Level Balancing method for the Flood Control category on the control point. The method uses the empty space available in the channel at the control point that is calculated by the regulation discharge method to balance the levels of associated reservoirs over a specified balance period. The empty space is specifically allocated to the reservoirs over the control point's forecast period. The method populates results into the Balance Level and Share slots on the control point and the Target Balance Level slot on each associated reservoir.

This method also computes a maximum flood control release for each reservoir under its control and stores the result in the reservoir's slot Max Flood Control Release. This value is used by the Operating Level Balancing flood control algorithm (see Computational Subbasin). The value is a flow, computed as

$$M = (RSTG / TTLEV) * MEORD$$

where

- RSTG is the reservoir's forecast storage at the end of the balance period above the target level, limited by the outflow control points regulation discharge,
- TTLEV is the total empty space volume at this control point at the end of the balance period, and
- MEORD is the maximum empty space hydrograph ordinate at this control point (over the forecast period), a flow.

Note, another way to think about this is that there are two ratios that are equal. The first is the maximum release over the total volume above the balance level. The second is the maximum empty space ordinate over the total empty space volume.

SLOTS SPECIFIC TO THIS METHOD

☞ KEY CONTROL POINT RESERVOIRS

Type: ListSlot

Units: NONE

Description: List of upstream reservoirs associated with this control point for balancing.

Information: Each reservoir in this list gets an associated column in the Share slot.

I/O: Required input

Links: Not linkable

☞ BALANCE PERIOD

Type: ScalarSlot

Units: NONE

Description: Single number representing the number of timesteps in the balance period.

Information: The balance period is the number of timesteps including the current simulation timestep that is used in the algorithm for balancing the reservoirs associated with a key control point. This value can be propagated from a computational subbasin of which the control point is a member.

I/O: Required input

Links: Not linkable

☞ BALANCE TOLERANCE

Type: ScalarSlot

Units: VOLUME

Description: In the iterative calculation for determining the balance level, balance tolerance allows the user to specify how close the storage above the balance level in the reservoirs at the end of the balance period must be compared to the empty space available in the channel over the balance period. Balance tolerance is expressed as a difference between these volumes.

Information: The algorithm to find a balance level where storage equals empty space will iterate until the balance tolerance is reached or until the number of balance iterations is reached. These parameters allows the user to control and limit the number of iterations of the algorithm.

I/O: Required input

Links: Not linkable

☞ Balance Iterations

Type: ScalarSlot

Units: NONE

Description: This parameter is how many times the balance level algorithm is allowed to iterate in determining a balance level where storage in the reservoirs above this level at the end of the balance period equals the empty space available in the channel over the balance period.

Information: The algorithm to find a balance level where storage equals empty space will iterate until the number of balance iterations is reached or until the balance tolerance is reached. These parameters allow the user to control and limit the number of iterations of the algorithm.

I/O: Required input

Links: Not linkable

☞ Balance Level

Type: SeriesSlot

Units: NONE

Description: Contains the result of the balance level determination for the key control point.

Information: The balance level calculated for the key control point is written to this Series Slot for each simulation timestep. This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool **HERE (USACE_SWD.pdf, Section 5.5)**.

I/O: Output

Links: Not linkable

☞ Share

Type: AggSeriesSlot

Units: DECIMAL

Description: Contains the results of allocating the empty space at the key control point to its associated reservoirs. Values are a proportional allocation and will add up to 1 for each timestep.

Information: Each column is a SeriesSlot and represents the allocated space for one of the reservoirs from the Key Control Point Reservoirs list. Space is allocated for the current simulation timestep through the forecast period. When the next timestep executes, the future forecast timestep values are overwritten by the new simulation and forecast values. This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool **HERE (USACE_SWD.pdf, Section 5.5)**.

I/O: Output

Links: Not linkable

The Operating Level Balancing method for the control point executes the following steps:

- Total the volume of empty space over the balance period at the control point.
- Find the total storage above conservation level summed for all associated reservoirs at the end of the balance period. Volume contributed by a reservoir is limited to the regulation discharge of its downstream gage control point at the current simulation timestep multiplied by the number of timesteps in the balance period.
- If there is no storage, or some storage, but not enough to fill the total empty space, set the balance level to the conservation pool level.
- Otherwise, determine the balance level at the end of the balance period where the total storage above that level equals the total empty space. The volume contributed by a reservoir is limited to the regulation discharge of its downstream gage control point at the current simulation timestep multiplied by the number of timesteps in the balance period. The bisection algorithm to determine balance level iterates until the Balance Tolerance for comparing storage to empty space is reached or until the Balance Iterations limit is reached.
- Assign the key control point's balance level to the associated reservoirs. If a reservoir is associated with more than one key control point, the highest balance level of the associated control points is used.
- Allocate the empty space at the control point to the reservoirs for each timestep in the forecast period. A reservoir's share for any forecast timestep is its volume above the balance level at the end of the balance period, provided at least part of a release from that reservoir on the current simulation timestep would reach the key control point by that forecast timestep, divided by the total volume above the balance level at the end of the balance period for all reservoirs whose releases would reach the key control point by that forecast timestep. Again the volume contributed by a reservoir is limited to the regulation discharge of its downstream gage control point for the current simulation timestep multiplied by the number of timesteps in the balance period.

9.1.14.2 Phase Balancing

This method is only available at the selection of the Phase Balancing method under the Flood Control category of the control point. It instantiates the phase space hydrograph of the control point's channel used by the Phase Balancing method on the Computation Subbasin.

The phase space hydrograph is used to allocate space to the reservoirs in a given phase over the control point's forecast period and a maximum flood control release value is calculated for each associated reservoir. (See Phase Balancing discussion in the Computational Object for a complete description).

Slots Specific to this method

☞ Phase Space Hydrograph

Type: TableSlot

Units: NONE, FLOW

Description: Contains flow capacity limits of each phase for this control point.

I/O: Required Input

Links: Not Linkable

☞ Phase Space Tolerance

Type: AggSeriesSlot

Units: FLOW

Description: Contains a series of tolerance values for defining each phase level of the control point.

Information: Each column (SeriesSlot) represents tolerance values for each phase level of the control point. The tolerance value is added to the phase space hydrograph to calculate the flow capacity limits of each phase for this control point.

I/O: Input

Links: Not Linkable

☞ Number Of Phases

Type: ScalarSlot

Units: NONE

Description: The number of phases associated with the all the phase space hydrograph.

Information: Value must agree with slot of same name on the computational subbasin.

I/O: Input

Links: Not linkable

9.1.15 Flooding Exception

This category appears at the selection of any regulation discharge method in the Regulation Discharge category, so is related to the operating level balancing approach to flood control. This category allows the user to specify reservoirs that are permitted to cause flooding at a control point.

9.1.15.1 None

This default method for the category means no flooding exception is specified.

9.1.15.2 Flooding Exception

This method associates with the control point a set of excepted reservoirs and an excepted flood release from each of those reservoirs. During execution of the Operating Level Balancing flood control method of the Flood Control category for a computational subbasin, the slots listed below are used in calculating a modified flood control release under the flooding exception part of the algorithm.

Slots Specific to this Method

☞ Excepted Reservoirs

Type: ListSlot

Units: NONE

Description: Contains a list of upstream reservoirs that are allowed to cause flooding at the control point.

Information: Each reservoir in this list automatically has a corresponding column created in the Excepted Flood Releases slot.

I/O: Required input

Links: Not linkable

☞ Excepted Flood Releases

Type: TableSlot

Units: FLOW

Description: Contains the excepted flood release values associated with the reservoirs in the Excepted Reservoirs slot.

Information: This is a 1 x n table slot where a column is created to match each reservoir in the Excepted Reservoirs slot. The one flow value entered for each reservoir represents its excepted amount of flood release for the control point. (This represents the amount of water that may be released in one timestep from the reservoir, not the amount that will arrive at the control point after routing.)

I/O: Required input

Links: Not linkable

9.1.16 Hydropower Flooding Exception

This category appears at the selection of any regulation discharge method in the Regulation Discharge category, so is related to the operating level balancing approach to flood control. This category allows the user to specify that upstream reservoirs are permitted to cause flooding with hydropower releases at this control point.

9.1.16.1 None

This default method for the category means no hydropower flooding exception is specified.

9.1.16.2 Releases Not Constrained by Flooding

This method specifies that flooding at the given control point will not constrain hydropower releases. This method is checked within execution of the HydropowerRelease predefined function HERE (RPLPredefinedFunctions.pdf, Section 96). If this method is selected, the computational subbasin will skip this control point when it is checking for additional downstream flooding at control points.

9.1.17 Low Flow Requirement

The Low Flow Requirement category is used to configure the Control Point to have a required low flow rate. In addition to computing the low flow requirement, these methods add slots that can be referenced by RPL logic (the MeetLowFlowRequirement function) to determine the reservoir releases required to meet the low flow rates.

9.1.17.1 None

This is the default method. There are no slots or calculations associated with it.

9.1.17.2 Low Flow Periodic Lookup

This method is used to specify the low flow requirement as a function of date/season using a periodic slot.

SLOTS SPECIFIC TO THIS METHOD

☞ **LOW FLOW REQUIREMENT**

Type: Periodic Slot

Units: FLOW

Description: The low flow requirement as a function of date/season

Information: The low flow requirement is computed at the beginning of each timestep using the current date to look up the value from this slot.

I/O: Required input

Links: Not linkable

☞ **COMPUTED LOW FLOW REQUIREMENT**

Type: Series Slot

Units: FLOW

Description: The computed low flow requirement

Information: This slot is computed given the current date and the Low Flow Requirement slot.

I/O: Output only

Links: Not linkable

☞ **LOW FLOW DEFICIENCY**

Type: Series Slot

Units: FLOW

Description: The portion of the low flow requirement that has not been met.
I/O: Output only
Links: Not linkable

☞ Low Flow Reservoirs

Type: List Slot
Units: NONE
Description: A list of the reservoirs that are used to meet the low flow requirement
Information: These reservoirs must all be upstream of the control point.
I/O: Required input
Links: Not linkable

The Low Flow Periodic Lookup method executes at the beginning of the run. Each timestep in the run is used to look up a value in the Low Flow Requirement slot. The resulting value will be set in the Computed Low Flow Requirement slot. The Low Flow Deficiency slot is computed each time the object dispatches. It is computed as the low flow requirement minus the total control point flow. The Low Flow Reservoirs slot is used by a RPL function (MeetLowFlowRequirement) to determine the reservoir releases necessary to meet the low flow requirement.

9.1.17.3 Reservoir Level Lookup

The Reservoir Level Lookup method is used to compute the low flow requirement as a function of the date/season and the operating level of a specified reservoir.

Slots Specific To This Method

☞ Level vs Low Flow Requirement

Type: Periodic Slot
Units: FLOW
Description: The low flow requirement as a function of reservoir level and date/season
Information: Each column corresponds to a reservoir's operating level. For each operating level, the requirements are specified for each date range in the periodic slot.
I/O: Required input
Links: Not linkable

☞ Low Flow Requirement Reservoir

Type: List Slot

Units: NONE

Description: The reservoir used to compute the low flow requirement

Information: The previous timestep operating level of this reservoir is used to look up the low flow requirement in the Level vs Low Flow Requirement slot.

I/O: Required input

Links: Not linkable

Computed Low Flow Requirement

Type: Series Slot

Units: FLOW

Description: The computed low flow requirement

Information: This slot is computed from the Level vs Low Flow Requirement slot.

I/O: Output only

Links: Not linkable

Low Flow Deficiency

Type: Series Slot

Units: FLOW

Description: The portion of the low flow requirement that has not been met.

I/O: Output only

Links: Not linkable

Low Flow Reservoirs

Type: List Slot

Units: NONE

Description: A list of the reservoirs that are used to meet the low flow requirement

Information: These reservoirs must all be upstream of the control point.

I/O: Required input

Links: Not linkable

Forecast Period

Type: TableSlot

Units: NONE

Description: The forecast period is a number of timesteps, including the current simulation timestep, that is used in the algorithms for calculating forecasted hydrology, regulation discharge and flood releases. This can be propagated from a computational subbasin of which the control point is a member.

Information:

I/O: Required input

Links: Not linkable

Note: If the specified reservoir is disabled and is set to Pass Inflows, no low flow requirement can be computed. See the note HERE (USACE_SWD.pdf, Section 4.3.4).

The Reservoir Level Lookup method will execute on the Control Point object at the beginning of each timestep. For each timestep (t) in the forecast period, the Low Flow Requirement Reservoir's previous timestep's (current timestep - 1) operating level and the forecast timestep's date (t) will be used to lookup a value in the Level vs Low Flow Requirement slot. The resulting value will be set in the Computed Low Flow Requirement slot. This is an estimated requirement that is used through the forecast period. It is reset at the beginning of the next timestep. Whenever the Control Point object dispatches, it will compute the Low Flow Deficiency based on the flow through the control point. The Low Flow Reservoirs slot will be used by the RPL pre-defined function to determine the low flow releases. The RPL function will order the reservoirs specified in the Low Flow Reservoirs slot in descending order beginning with the most full (highest level) reservoir. Each reservoir makes a release until the requirement is met or the reservoir reaches its maximum low flow delivery rate.

9.1.18 Instream Flow Reference Level

This category is used in conjunction with water accounting, specifically for water rights allocation, to establish a reference level from which an instream flow account on this object can base its water allocation requests.

The selected method is executed at the beginning of each timestep.

9.1.18.1 None

This default method for the category means no reference level is to be computed.

9.1.18.2 Reservoir Storages Lookup

This method executes one timestep each year identified by the Start of Reference Year slot. At that timestep, this method sums the storages at the previous timestep from each of the Reference Reservoirs. This storage is then looked up in the Reference Level Storage Table to determine the Reference Level. The Reference Level is used by the instream flow accounts on the object to determine the account's Initial Request.

Slots Specific to this Method

☞ Reference Reservoirs

Type: ListSlot

Units: NONE

Description: Contains a list of reservoirs whose storages (at the previous timestep) will be summed.

Information: This slot is used by instream flow accounts on the object for water rights allocation. Its value is an input to the calculation of Initial Requests for allocation of water.

I/O: Required input

Links: Not linkable

☞ Reference Level Storage Table

Type: TableSlot

Units: VOLUME vs NONE

Description: Maps volumes (sum of storages) to a reference level. Contains two columns, "Sum of Storages" and "Reference Level". The user may add any number of rows, but at least one row must be present and populated.

Information: The sum of storages is looked up in the first column and the corresponding value in the second column is put into the "Reference Level" slot. No interpolation is done. If the sum falls between two values in the first column, the row chosen is the row with the largest first-column value that is less than the sum.

I/O: Required input

Links: Not linkable

☞ Reference Level

Type: Series Slot with optional annual timestep

Units: NONE

Description: The resulting reference level is placed here. This slot can be set to an annual-timestep slot; its value is computed once per annum, on the timestep identified by the Start of Reference Year slot. Because annual-timestep slots in RiverWare are given the December 31 date, the date of the timesteps will not match the Start of Reference Year (unless you set the start of the year to December 31!). If it is not an annual timestep slot and the slot's timestep matches the run's timestep, the value is computed each timestep in the run.

Information: This slot is used by the instream flow accounts on this object. The value in this slot is used by a method that calculates the account's Initial Request slot. On timesteps before the Start of Reference Year, the account method uses the previous year's Reference Level, so you must populate this slot with an initial value if the Start of Reference Year does not coincide with the first timestep of the run.

I/O: Output

Links: Not linkable

☞ Start of Reference Year

Type: TableSlot

Units: NONE, NONE

Description: Identifies month and day of timestep that is the start of the reference year; the timestep on which a new Reference Level will be computed.

Information: Contains two columns, "Month" and "Day". Only the first row is used. See the information for the Reference Level slot, above for more information. Note, the reference year cannot start on February in a monthly model.

I/O: Required input

Links: Not linkable

9.1.19 Variable Routing Coefficients

This category allows the user to specify that a given control point should be part of the alternative routing coefficients computation. This category is dependent on the Operating Level Balancing in the Flood Control category.

9.1.19.1 None

This is the default method. The existing Routing Coefficients slot is used for all routing.

9.1.19.2 Compute Aggregate Coefficients

The Compute Aggregate Coefficients method recalculates the routing coefficients for each upstream reservoir (where necessary) using intervening Reach routing. This allows the Flood Control algorithm to make use of the alternative routing coefficients that are to be used at higher flow rates. The bulk of this calculation is performed by the computational subbasin and is described in the Computation Subbasin's Compute Aggregate Coefficients documentation, HERE (Section 7.1.24.2).

For this method, you must also specify the "standard" coefficients on the Routing Coefficients slot.

Slots Specific to this Method

☞ Computed Routing Coefficients

Type: TableSlot

Units: No Units

Description: This is a temporary table slot (i.e. it is not saved with the model file) used to store the routing coefficients used on the current timestep. Values in the table are populated at the beginning of each timestep, as necessary, based on the flows from the upstream reservoir to the Control Point. This is described further on the Computation Subbasin's Compute Aggregate Coefficients method.

Information: There is a column for each reservoir in the upstream reservoir list. At beginning of run, the structure of this table is copied from the Routing Coefficients slot. Only coefficients in the forecast period are used, so any values past the end of the forecast period are zero. Therefore, the columns do not necessarily sum to 1.0.

I/O: Output only and read-only.

Links: No

This method builds the Computed Routing Coefficients slot which is used by the computational subbasin to store the computed alternative coefficients at the beginning of the timestep. During the FloodControl function execution, if the control point has this method selected, the routing coefficients in the Computed Routing Coefficients slot will be used for all computations.

9.1.19.3 Compute Aggregate Coeffs every Timestep

The Compute Aggregate Coefficients every Timestep method recalculates the routing coefficients for each upstream reservoir at the start of **every** timestep using intervening Reach routing. This allows the Flood Control algorithm to make use of the alternative routing coefficients that are to be used at higher flow rates. The bulk of this calculation is performed by the computational subbasin and is described in the Computation Subbasin's Compute Aggregate Coefficients documentation, HERE (Section 7.1.24.2).

For this method, there is no need to specify the "standard" coefficients on the Routing Coefficients slot. The slot will always exist, but all values can be NaN.

Slots Specific to this Method

☞ Computed Routing Coefficients

Type: TableSlot

Units: No Units

Description: This is a temporary table slot (i.e. it is not saved with the model file) used to store the routing coefficients used on the current timestep. Values in the table are populated at the beginning of each timestep based on the flows from the upstream reservoir to the Control Point. This is described further on the Computation Subbasin's Compute Aggregate Coefficients method.

Information: There is a column for each reservoir in the upstream reservoir list. At beginning of run, the structure of this table is copied from the Routing Coefficients slot. Only coefficients in the forecast period are used, so any values past the end of the forecast period are zero. Therefore, the columns do not necessarily sum to 1.0.

I/O: Output only and read-only.

Links: No

This method builds the Computed Routing Coefficients slot which is used by the computational subbasin to store the computed alternative coefficients at the beginning of the timestep. During the FloodControl function execution, if the control point has this method selected, the routing coefficients in the Computed Routing Coefficients slot will be used for all computations.

9.2 Dispatch Methods

This object has the following dispatch method:

9.2.1 solveOutflowGivenInflow

The required knowns and unknowns (at the current timestep) are listed below.

Required Knowns

☞ **Inflow**

Required Unknowns

☞ **Outflow**

The dispatch method solves for the outflow from the Control Point using known values for Inflow and Local Inflow; however, a Local Inflow value is not required for the object to dispatch. If there is no Local Inflow, the Outflow is simply equal to the Inflow. If there is a Local Inflow value, the Outflow is the sum of the Inflow and Local Inflow.

It is also possible that the method Locals Not Included in Outflow is turned on. In this case a new slot, Total Discharge, is available and is the sum of the Inflow and the Local Inflow.

$$TotalDischarge = Inflow + LocalInflow$$

With this method turned on the Local Inflow is not included in the Outflow and therefore Outflow is again simply equal to the Inflow.

$$Outflow = Inflow$$

Note: Flow is not routed through the Control Point. That is, the Routing Coefficients slot values are not used in the solution of Control Point Outflow. They are strictly used in flood control and other basin wide operations described HERE (Section 9.1.8).

If the regulation discharge (G) flag is set for the timestep on the Reg Discharge Calculation slot, the dispatch method recognizes this and executes the selected regulation discharge method and any dependent methods for the control point. This flag can only be set by a rule. After execution, the regulation discharge flag is removed so that regulation discharge will not redispatch unless this flag is reset by a rule. Each time regulation discharge is executed for a timestep, the value in the Reg Discharge Calculation slot is incremented so that it records the number of times that regulation discharge has been executed for the timestep. A rule (R) flag will typically remain in this slot after execution to show that its value was set as a result of a rule.

After executing regulation discharge, the dispatch method checks to see if outflow has changed from its previous value. If it has not, the outflow slot is not reset using the regulation discharge rule priority. This prevents the regulation discharge rule priority from propagating downstream and triggering unnecessary and, in some cases, undesirable resolving.

10. Data Object

The data object provides a container to hold user data that is not appropriate for simulation objects. The data object can be used to hold many types of data including series, table, periodic, scalar, expression, and statistical. There are neither general slots nor methods on a data object.

Data objects are empty when first created. You populate them with custom slots based on your needs. You can create various types of slots on the data object as described HERE (ObjectDialogs.pdf, Section 2.2.4).

11. Distribution Canal

The Distribution Canal is an element within an AggDistributionCanal object. It routes its Delivery Requests and any downstream delivery requests to the upstream end of the element.

Links among Distribution Canal elements and between Distribution Canal elements and the AggDistributionCanal object are maintained by **RiverWare**™. Automatic links created by **RiverWare**™ are listed below:

The Total Inflow slot on the AggDistributionCanal object is linked to the Inflow slot on the first Distribution Canal element.

The Outflow slot of all but the last canal element is linked to the Inflow slot of the next canal element.

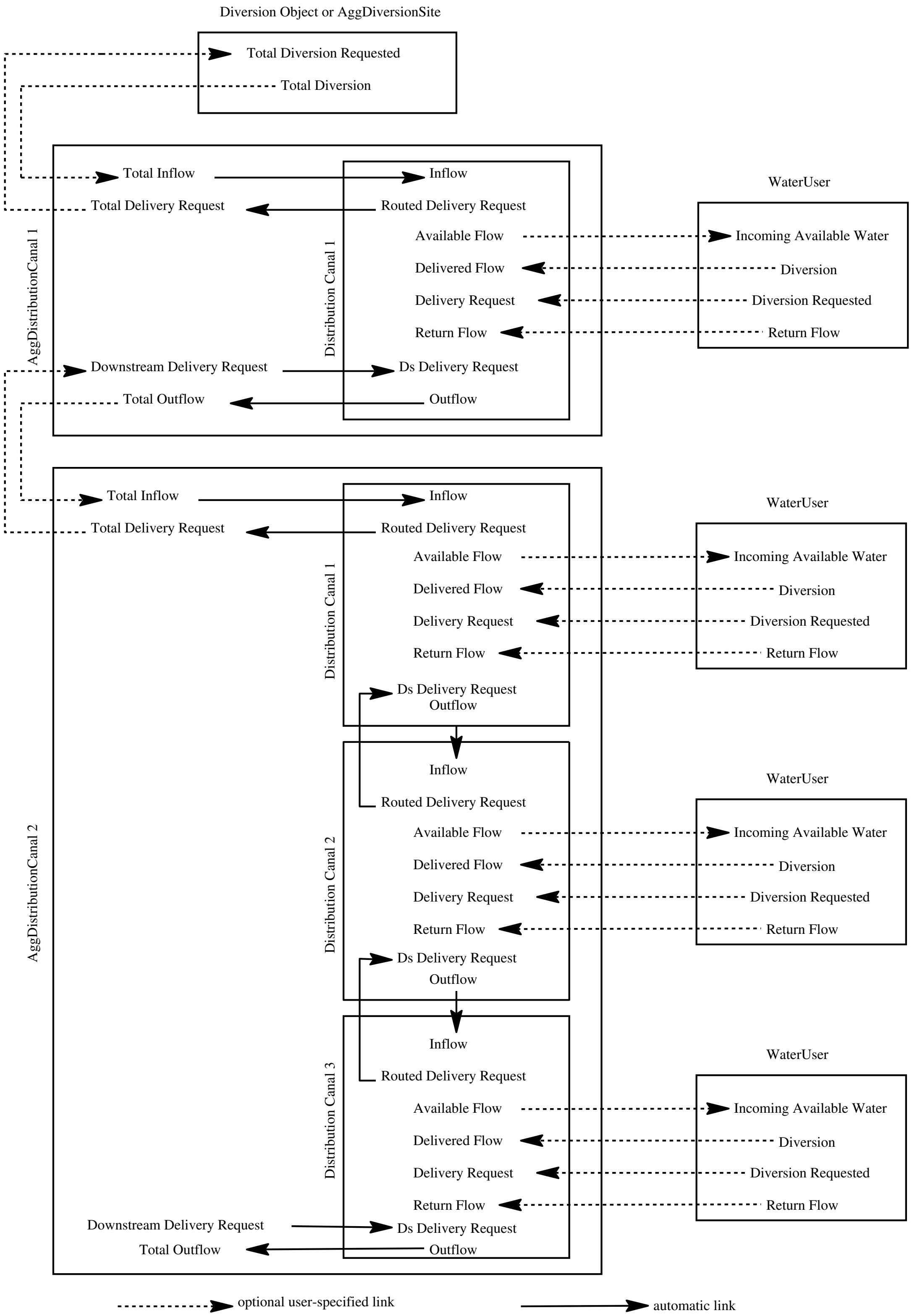
The last distribution canal element's Outflow slot is linked to the Total Outflow slot on the AggDistributionCanal object.

The Routed Delivery Request slot on all but the first Distribution Canal element is linked to the DS Delivery Request slot of the preceding Canal element.

The Routed Delivery Request slot of the first Distribution Canal element is linked to the Total Delivery Request slot of the AggDistributionCanal object.

The Downstream Delivery Request slot on the AggDistributionCanal object is linked to the DS Delivery Request slot on the last Distribution Canal element.

The diagram below shows the linking structure just described. The automatic links are represented by a solid line while optional user-specified links are shown with a dashed line. The example shows how two AggDistributionCanal objects can interact with each other and how they each interact with separate WaterUsers. WaterUsers can be elements on the Agg Diversion Site when the No Structure linking structure is used, or they can exist as separate objects on the workspace.


Diversion Object or AggDiversionSite
Total Diversion Requested
Total Diversion
AggDistributionCanal 1
Total Inflow
Total Delivery Request
Downstream Delivery Request
Total Outflow
Distribution Canal 1
Inflow
Routed Delivery Request
Available Flow
Delivered Flow
Delivery Request
Return Flow
Ds Delivery Request
Outflow
WaterUser
Incoming Available Water
Diversion
Diversion Requested
Return Flow
AggDistributionCanal 2
Total Inflow
Total Delivery Request
Distribution Canal 1
Distribution Canal 2
Distribution Canal 3
Downstream Delivery Request
Total Outflow
optional user-specified link
automatic link

General Slots

(slots which always appear for this object)

☞ Available Flow

Type: SeriesSlot
Units: FLOW
Description: flow at the downstream end of the Distribution Canal element
Information: Available Flow is the flow that may be diverted. It is calculated by the routing method, and takes into account any gain/loss or lag time that occurs in the Distribution Canal element.
I/O: Output only
Links: Usually linked to the Incoming Available Water slot on a WaterUser or the Total Available Water slot on an Agg Diversion Site.

☞ Delivered Flow

Type: SeriesSlot
Units: FLOW
Description: flow delivered to the WaterUser
Information: This slot is usually linked to the Diversion slot in a WaterUser. If there is no link, all non-input values are set to zero.
I/O: Optional; usually set from a link
Links: Usually linked to the Diversion slot on the diverting object.

☞ Inflow

Type: SeriesSlot
Units: FLOW
Description: inflow to the Distribution Canal element
Information:
I/O: Output only
Links: The Inflow slot can be set via link propagation from the Total Inflow slot on the AggDistributionCanal object (if it is the first Canal element) or from the Outflow slot of the previous Distribution Canal element.

☞ Outflow

Type: SeriesSlot
Units: FLOW
Description: outflow from the Distribution Canal element
Information: Outflow is calculated by the dispatch method.
I/O: Output only
Links: Linked to the Inflow slot of the next Distribution Canal element or to the Total Outflow slot on the AggDistributionCanal if it is the last element.

☞ Return Flow

Type: SeriesSlot
Units: FLOW
Description: return flow to downstream end of the Distribution Canal element
Information: Usually linked to the Return Flow slot of the Water User. If it is not linked, any non-input values will be set to 0.
I/O: Optional; usually set by propagation across a link
Links: May be linked to the Return Flow slot on a WaterUser, the Total Return Flow slot on an Agg Diversion Site, or the Total Unused Water slot on an Agg Diversion Site

11.1 User Methods

11.1.1 Canal Max Capacity

The Canal Max Capacity category is used to limit the amount of flow at the top of the Distribution Canal element. This category simply turns the application of maximum capacity on and off.

11.1.1.1 None

This method does not limit the flow through the canal. It performs no calculations. There are no slots specifically associated with this method.

11.1.1.2 Input Max Capacity

This method limits the flow into the top of the canal to a user specified amount. Any flow that exceeds the Maximum Capacity is spilled from the canal.

SLOTS SPECIFIC TO THIS METHOD

☞ **CANAL SPILLOVER**

Type: SeriesSlot
Units: FLOW
Description: flow spilled from the canal because the maximum capacity was exceeded.
Information: Any inflow into the canal that exceeds the maximum capacity will be placed in the slot, and the maximum capacity of the canal will then be routed.
I/O: Output only
Links: May be linked to the Inflow slot on a Groundwater Storage object or to a Reach object.

☞ **MAXIMUM CAPACITY**

Type: ScalarSlot
Units: FLOW
Description: maximum amount of flow that can enter the upstream end of the canal
Information:
I/O: Required input
Links: Not linkable

When flow is being routed downstream, Canal Spillover is calculated as follows (if Inflow is greater than Maximum Capacity):

$$CanalSpillover = Inflow - MaximumCapacity$$

The Maximum Capacity is then routed downstream.

When the requests are being routed upstream, this method ensures that the requests do not exceed the Maximum Capacity.

11.1.2 Canal Seepage

The Canal Seepage method category is used to model seepage from the canal. This category controls how to apply the seepage.

11.1.2.1 None

This is the default method. It performs no calculations. There are no slots specifically associated with this method.

11.1.2.2 Input Seepage

This method allows users to directly input values into the Seepage slot by hand or to set the slot with a rule.

☞ **CANAL SEEPAGE**

Type: SeriesSlot
Units: FLOW
Description: the calculated amount of flow lost as seepage
Information:
I/O: Input only
Links: May be linked to the Inflow slot on a Groundwater Storage object or to a series slot on the Data Object.

11.1.2.3 Proportional Seepage

This method uses constant parameters to model seepage. Seepage is lost from the canal after any flow that exceeds the Maximum Capacity is spilled, but before the flow is routed.

SLOTS SPECIFIC TO THIS METHOD

☞ **CANAL SEEPAGE**

Type: SeriesSlot
Units: FLOW
Description: the calculated amount of flow lost as seepage
Information:
I/O: Output only
Links: May be linked to the Inflow slot on a Groundwater Storage object.

☞ **MAXIMUM SEEPAGE**

Type: SeriesSlot
Units: FLOW
Description: represents the maximum amount of flow that can be lost as seepage

Information:
I/O: Optional; this slot is not used if it is not input.
Links: Not linkable

☞ **SEEPAGE FLOW FRACTION**

Type: ScalarSlot
Units: NO UNITS
Description: represents the fraction of Inflow that is lost as seepage
Information: Entered as a decimal value which is greater that or equal to zero but less than one.
I/O: Optional; set to zero if not input.
Links: Not linkable

Canal Seepage is calculated as:

$$\text{Canal Seepage} = MIN(\text{Maximum Seepage, Inflow} \times \text{Seepage Flow Fraction})$$

Canal Seepage is then subtracted from the Inflow before the flow is routed.

Canal Seepage is accounted for when routing requests upstream (it is added it to the request).

11.1.2.4 Variable Seepage

This method uses variable parameters to model seepage. Seepage is lost from the canal after any flow that exceeds the Maximum Capacity is spilled, but before the flow is routed.

SLOTS SPECIFIC TO THIS METHOD

☞ **CANAL SEEPAGE**

Type: SeriesSlot
Units: FLOW
Description: the calculated amount of flow lost as seepage
Information:
I/O: Output only
Links: May be linked to the Inflow slot on a Groundwater Storage object.

☞ **MAXIMUM SEEPAGE**

Type: SeriesSlot
Units: FLOW
Description: represents the maximum amount of flow that can be lost as seepage
Information:
I/O: Optional; this slot is not used if its not input.
Links: Not linkable

☞ **VARIABLE SEEPAGE FLOW FRACTION**

Type: SeriesSlot
Units: NO UNITS
Description: a time series of values representing the fraction of Inflow that is lost as seepage
Information: Entered as a decimal value which is greater than or equal to zero but less than one.
I/O: Optional; set to zero if not input.
Links: Usually not linked

Canal Seepage is calculated as:

$$\text{Canal Seepage} = MIN(\text{Maximum Seepage}, \text{Inflow} \times \text{Variable Seepage Flow Fraction})$$

Canal Seepage is then subtracted from the Inflow before the flow is routed.

Canal Seepage is accounted for when routing requests upstream (it is added it to the request).

11.1.2.5 Head Based Seepage

The Head Based Seepage method, based on the method of the same name for the reach object, calculates seepage based on the elevation of the canal water relative to the groundwater elevation. If the groundwater head is below the canal bed elevation then the seepage will be based on the difference between the canal water surface elevation and the canal bed elevation. Note that all elevations should share the same datum. The following slots and calculations will be associated with this method:

☞ **CANAL WATER SURFACE ELEVATION**

Type: TableSlot
Units: LENGTH
Description: Average elevation of the water in the canal
Information: Should have the same datum as stage and the previous water table elevation
I/O: Required input
Links: Not linkable

☞ **SEEPAGE**

Type: Series Slot
Units: FLOW
Description: The canal loss to groundwater
Information: A positive number represents a losing reach while a negative number represents a gaining reach
I/O: Output only
Links: Usually linked to the groundwater object

☞ Previous Water Table Elevation

Type: Series Slot
Units: LENGTH
Description: The previous elevation value computed by the connected groundwater object
Information: Seepage is computed based on the difference between the current canal water surface elevation and the previous water table elevation
I/O: Output only
Links: Linked to the Previous Elevation slot on the connected groundwater object

☞ Conductance

Type: Scalar slot
Units: AREA PER TIME
Description: The riverbed conductance
Information: The conductance is defined as the hydraulic conductivity of the streambed material, multiplied by the width of the streambed bottom, multiplied by the length of the stream segment, divided by the streambed thickness (Conductance = KwL/m).
I/O: Input or computed as specified HERE (Section 11.1.4).
Links: Not linkable

☞ Canal Bed Elevation

Information: Scalar Slot
Units: LENGTH
Description: The elevation of the canal bed at the center of the canal segment (i.e. average canal bed elevation)
Information: Should have the same datum as Canal Water Surface Elevation and the Previous Water Table Elevation
I/O: Required input
Links: Not linkable

The method will compute the seepage to groundwater as follows:

If the Previous Water Table Elevation is greater than the Canal Bed Elevation:

$$\text{Seepage} = \text{Conductance} \times (\text{Canal Water Surface Elevation} - \text{Previous Water Table Elevation})$$

If the water table elevation at the previous timestep is below the Streambed Elevation:

$$\text{Conductance} \times (\text{Canal Water Surface Elevation} - \text{Canal Bed Elevation})$$

Note that the equations above use the water table elevation at the previous timestep. This simplification is done to avoid iteration problems that would result if the current groundwater elevation was used. A positive seepage is limited to be less than or equal to the Inflow to the canal.

11.1.3 Canal Seepage Routing

The option to route seepage becomes visible after a method to calculate seepage is chosen. In general, a seepage routing method makes visible a PreRouted seepage slot. This slot contains the values for seepage that have not been routed. The routed values are contained in the Canal Seepage slot. If a link exists that links seepage to another object and a seepage routing method is chosen, the link will propagate the values of the routed seepage.

11.1.3.1 No Routing

This is the default method which will not route seepage. The values for seepage are calculated depending on the user method chosen for Canal Seepage. These values are contained in the Canal Seepage slot.

11.1.3.2 Impulse Response

This method will calculate the routed seepage based on the impulse response method of routing.

SLOTS SPECIFIC TO THIS METHOD

☞ **NUMBER OF COEFFICIENTS**

Type: TableSlot
Units: NONE
Description: The integer number of lag coefficients to be used in the method.
I/O: Required input
Links: Not linkable

☞ **LAG COEFFICIENTS**

Type: TableSlot
Units: NONE
Description: The impulse response lag coefficients
Information: The number of lag coefficients must be the same as the value in the Number of Coefficients slot. The input will be in rows.
I/O: Required input.
Links: Not linkable.

☞ **PREROUTED SEEPAGE**

Type: SeriesSlot
Units: FLOW
Description: The seepage values before routing.
Information: This slot contains the values of seepage that have not been routed. The routed values will be contained in the Canal Seepage slot. If the user-method for Canal Seepage is Input Seepage, the values are to be inputted in this slot.

I/O: Output/Input (see Information)
Links: Can be linked, but slot containing routed values is Canal Seepage. If routed values are to be linked, then this slot should not be linked.

The Routed Seepage will be calculated as:

$$\text{RoutedSeepage}_t = C_0\text{PreRoutedSeepage}_t + C_1\text{PreRoutedSeepage}_{t-1} + \ldots.$$
$$+ C_{ncoeff-1}\text{PreRoutedSeepage}_{t-ncoeff-1} + C_{ncoeff}\text{PreRoutedSeepage}_{t-ncoeff}$$

where C is a lag coefficient and *ncoeff* is the number of lag coefficients.

11.1.4 Canal Conductance Specification

Methods in this category allow you to choose how you wish to specify conductance. You can either give values for Conductance or give values for Hydraulic Conductivity and the geometry of the reach and the Conductance will be computed.

11.1.4.1 Specify Conductance

This is the default method and does not instantiate any new slots.

11.1.4.2 Compute Conductance

This method allows you to specify hydraulic conductivity and the geometry.

THE FOLLOWING SLOTS ARE ADDED:

☞ **HYDRAULIC CONDUCTIVITY**

Type: Scalar
Units: VELOCITY
Description: Hydraulic conductivity of the bed material
Information:
I/O: Required Input
Links: Not Linkable

☞ **CANAL SEEPAGE THICKNESS**

Type: Scalar
Units: LENGTH
Description: Thickness associated with the conductance calculation.
Information:
I/O: Required Input
Links: Not Linkable

☞ **SEEPAGE AREA**

Type: Scalar
Units: AREA
Description: Area of the river that contributes to seepage
Information:
I/O: Required Input
Links: Not Linkable

At the start of the run, the information in these slots is used to compute the **Conductance** value according to the following equation:

$$\text{Conductance} = \text{Hydraulic Conductivity} \times \frac{\text{Seepage Area}}{\text{Riverbed Thickness}}$$

If there are missing values in the above slots, an error will be issued and the run initialization will be stopped.

The resulting value is set on the scalar **Conductance** slot. If there are values already in the slot, they will be overwritten.

Note: The **Conductance** slot is registered as having a “Source” slot. When a slot has a source slot, it becomes read-only and displays a cross hatch over the data. It also provides a note indicating the source used to compute the data. This attribute is set at the start of the run, so you must initialize the model to see the display of this slot change. If you deselect this method, you must initialize the run again to un-set the source slot.

11.1.5 Flow Routing

The Flow Routing method category is used to route the flow through the Distribution Canal element. If Time Lag Request Routing is selected on the containing AggDistributionCanal, only Time Lag Routing will be available in this category.

11.1.5.1 No Routing

This method sets the available flow to equal the inflow after subtracting Canal Spillover and/or Canal Seepage if applicable. There are no slots specifically associated with this method.

Requests cannot be routed upstream when this method is selected. Therefore, this method is only available if No Routing is selected in the Request Routing category on the containing AggDistributionCanal.

11.1.5.2 Time Lag

Routes flow through the Distribution Canal element using a time lagged approach. Canal Spillover and/or Canal Seepage are subtracted from the Inflow before routing. Requests can be routed upstream when this method is selected only if Time Lag routing is selected in the Request Routing category on the containing AggDistributionCanal.

Slots Specific to this Method

☞ **Delivery Request**

Type: SeriesSlot
Units: FLOW
Description: delivery request at the downstream end of the Canal element
Information:
I/O: Optional; Only used if routing requests.
Links: Usually linked to the Diversion Requested slot on a Water User or Total Diversion Requested on an Agg Diversion Site.

☞ **Ds Delivery Request**

Type: SeriesSlot
Units: FLOW
Description: delivery requests from any downstream objects
Information: The DS Delivery Request is added to the Delivery Request and the sum is routed to the upstream end of the Distribution Canal element. Only used if routing requests.
I/O: Optional; usually set by propagation across a link
Links: DS Delivery Request is either automatically linked to the Routed Delivery Request slot on the downstream element, or (if there are no downstream

elements) to the Downstream Delivery Request slot on the AggDistributionCanal object.

☞ **LAG TIME**

Type: TableSlot
Units: TIME
Description: lag time of the flow through the Distribution Canal element
Information: This must be an integer number of timesteps.
I/O: Optional; if not input, it is set to zero.
Links: Not linkable

☞ **ROUTED DELIVERY REQUEST**

Type: SeriesSlot
Units: FLOW
Description: routed delivery request at the upstream end of the Distribution Canal element
Information: The Routed Delivery Request is the sum of the DS Delivery Request and the Delivery Request, routed to the upstream end of the element. This slot is automatically linked, to either the DS Delivery Request of the upstream Canal element or to the Total Delivery Request on the AggDistributionCanal if there is no upstream element. Only used if routing requests. The Routed Delivery Request is always zero when the canal is draining (this only applies when a method is selected in the Canal Storage category).
I/O: Output only
Links: Automatically linked

The Time Lag method is used to compute Available Flow. The necessary calculations are given below:

$$\text{Lag Integer} = \frac{\text{Lag Time}}{\text{Time Step}}$$

The value of Available Flow is the Inflow, after subtracting Canal Spillover and/or Canal Seepage. It is set at the timestep, t + Lag Integer, where t is the current timestep.

11.1.5.3 Storage Time Routing

This method is a simple storage method that solves for Available Flow given current and previous Inflow values. The reach is broken up into a user-specified number of linked segments and flows are calculated for each segment. This method is almost identical to the Storage Routing method on the Reach object.

SLOTS SPECIFIC TO THIS METHOD

☞ Number of Segments in Reach

Type: ScalarSlot
Units: NOUNITS
Description: number of segments upstream to downstream
Information: This will determine the number of columns in the Segment Outflow table as well.
I/O: Required input
Links: Not Linkable

☞ Segment Outflow

Type: TableSeriesSlot
Units: FLOW
Description: segment outflow
Information: Available Flow is set as the outflow the last segment. The columns in this table will be resized to the number of segments input by the user, and the number of rows will be the number of timesteps in the run.
I/O: Output only
Links: Not linkable

☞ Storage Time Coefficient

Type: ScalarSlot
Units: NOUNITS
Description: value that is divided by the result of the average flow and exponent to arrive at time in storage
Information: The units of this slot should be $\text{Volume}^{\text{exponent}}$ (a value should be used that is in $(ft^3)^{\text{exponent}}$). This coefficient may be determined by trial and error, and should not be negative. The value must correspond to a flow value in cfs and storage time in hours, regardless of the user units on other slots.
I/O: Required input
Links: Not linkable

☞ Storage Time Exponent

Type: ScalarSlot
Units: NOUNITS
Description: exponent of mean flow value.
Information: Usually between -1 and 1. The value must correspond to a flow value in cfs and storage time in hours, regardless of the user units on other slots.
I/O: Required input
Links: Not linkable

The algorithm proceeds in this fashion:

- If the previous Inflow value is not known, the method exits.

- The outflow value for each segment from the previous timestep is checked for validity. There are then three possible scenarios:

  1. If the segment outflows are not valid and previous Available Flow is not valid, Available Flow is set equal to Inflow plus gain loss and the method exits.

  2. If the segment outflows are not valid, and previous Available Flow is valid, set all segment outflows equal to previous Inflow, and continue the routing method.

  3. If the segment outflows are valid, continue the method.

- Find the mean interior flow from the previous timestep as the average of all segment outflows.

- Find the time of storage in the reach based on the following empirical formula (in cfs):

$$storagetime = \frac{coefficient}{(meanfl)^{exponent}}$$

- where storagetime is the time of storage *in hours*, coefficient & exponent are user input constants in the slots above, and meanfl is the average interior flow of the previous timestep. (The storagetime calculation is always made with meanfl in units of cfs regardless of the user units on any flow slots. The values for Storage Time Coefficient and Storage Time Exponent should be set accordingly.) The time in storage can be used as a "conversion" from storage to outflow.

$$Storage = storagetime \bullet Outflow$$

- Find the number of routing phases, n. If the time of storage is greater than half of the simulation timestep, n is 1. Otherwise, n is calculated as:

$$n = \frac{timestep}{2storagetime} + 1$$

- If n is greater than 48 from this equation, n is set to 6.

- The inflow into the first segment for each routing phase is

$$segInflow = \frac{inflow + inflow(-1)}{2n}$$

- For each routing phase, the outflow from each segment is:

$$segOutflow = \frac{(segInflow - segOutflow(-1))timestep}{\frac{timestep}{2} + storagetime} + segOutflow(-1)$$

- This routing equation is based on the storage-outflow relation, storagetime, from above, and the continuity equation:.

$$\left(\frac{Inflow + Inflow(-1)}{2} - \frac{Outflow + Outflow(-1)}{2}\right)timestep = Storage - Storage(-1)$$

- The inflow into the next segment is the average of the segments previous and current outflows

The current timestep’s Available Flow for the object is equal to the last segment’s outflow.

11.1.5.4 Variable Time Storage Routing

This method is a simple storage method that solves for Available Flow given current and previous Inflow values. The reach is broken up into a user-specified number of linked segments and flows are calculated for each segment. This method differs from the Storage Routing method only in the determination of the storagetime value. In this method, the storage time exponent and coefficient are found from a table lookup based on flow instead of constant values. This method is almost identical to the Variable Storage Routing method on the Reach object.

Slots Specific to this Method

☞ Flow Range

Type: TableSlot
Units: FLOW
Description: the ranges of flow rates corresponding to the Variable Storage Time Tables
Information: The number of flow ranges allowed in this table is currently limited to 10. The columns of this table represent, in ascending order, the maximum flow rate for a specific flow range starting with 0 as the base. These columns are labeled **Flow Range 1 - 10**. Therefore, the first column represents a range of flows from 0 to the value input by the user in the **Flow Range 1** column. The last flow range entered by the user should be a flow rate greater than any anticipated for the simulation. Otherwise, a value of 0.0 will be used for the **Variable Storage Time** slots for flows outside of this range. It is not necessary to utilize all 10 columns of this table. Use only the columns needed to designate the desired flow ranges.
I/O: Required input
Links: Not linkable

☞ Number of Segments in Reach

Type: ScalarSlot
Units: NOUNITS
Description: number of segments upstream to downstream

Information: This will determine the number of columns in the Segment Outflow table as well.
I/O: Required input
Links: Not linkable

☞ Segment Outflow
Type: TableSeriesSlot
Units: FLOW
Description: segment outflow
Information: Available Flow is set as the outflow from the last segment. The columns in this table will be resized to the number of segments input by the user, and the number of rows will be the number of timesteps in the run.
I/O: Output only
Links: Not linkable

☞ Variable Storage Time Coefficient
Type: SeriesSlot
Units: NOUNITS
Description: value that is divided by the result of the average flow and exponent to arrive at time in storage
Information: The units of this slot should be $Volume^{exponent}$ (a value should be used that is in $(ft^3)^{exponent}$).
I/O: Output only
Links: Not Linkable

☞ Variable Storage Time Coefficient Table
Type: Tableslot
Units: NOUNITS
Description: Variable Storage Time Coefficient for each specific Flow Range
Information: The columns of the table correspond to the flow ranges defined in **Flow Range**. The values must correspond to a flow value in cfs and storage time in hours, regardless of the user units on other slots.
I/O: Required input
Links: Not linkable

☞ Variable Storage Time Exponent
Type: SeriesSlot
Units: NOUNITS
Description: exponent of mean flow value.
Information: Usually between -1 and 1.
I/O: Output only
Links: Not linkable

☞ Variable Storage Time Exponent Table

Type: Tableslot
Units: NOUNITS
Description: Lag times for each specific Flow Range
Information: The columns of the table correspond to the flow ranges defined in **Flow Range.** The values must correspond to a flow value in cfs and storage time in hours, regardless of the user units on other slots.
I/O: Required input
Links: Not linkable

The algorithm proceeds in this fashion:

- If the previous Inflow value is not known, the method exits.
- The outflow value for each segment from the previous timestep is checked for validity. There are then three possible scenarios:

  1. If the segment outflows are not valid and previous Available Flow is not valid, Available Flow is set equal to Inflow plus gain loss and the method exits.
  2. If the segment outflows are not valid, and previous Available Flow is valid, set all segment outflows equal to previous Inflow, and continue the routing method.
  3. If the segment outflows are valid, continue the method.

- Find the mean interior flow from the previous timestep as the average of all segment outflows.
- Find the time of storage in the reach based on the following empirical formula (in cfs):

$$storagetime = \frac{coefficient}{(meanfl)^{exponent}}$$

- where storagetime is the time of storage *in hours*, coefficient & exponent are user input constants in the slots above, and meanfl is the average interior flow of the previous timestep. (The storagetime calculation is always made with meanfl in units of cfs regardless of the user units on any flow slots. The values in the Variable Storage Time Coefficient Table and Variable Storage Time Exponent Table should be set accordingly.) The time in storage can be used as a “conversion” from storage to outflow:

$$Storage = storagetime \bullet Outflow$$

- Find the number of routing phases, n. If the time of storage is greater than half of the simulation timestep, n is 1. Otherwise, n is calculated as:

$$n = \frac{timestep}{2storagetime} + 1$$

- If n is greater than 48 from this equation, n is set to 6.
- The inflow into the first segment for each routing phase is

$$segInflow = \frac{inflow + inflow(-1)}{2n}$$

- For each routing phase, the outflow from each segment is:

$$segOutflow = \frac{(segInflow - segOutflow(-1))timestep}{\frac{timestep}{2} + storagetime} + segOutflow(-1)$$

- This routing equation is based on the storage-outflow relation, storagetime, from above, and the continuity equation:.

$$\left(\frac{Inflow + Inflow(-1)}{2} - \frac{Outflow + Outflow(-1)}{2}\right)timestep = Storage - Storage(-1)$$

- The inflow into the next segment is the average of the segments previous and current outflows

The current timestep's Available Flow for the object is equal to the last segment's outflow.

11.1.6 Canal Storage

This category is used to track canal storage.

11.1.6.1 None

This is the default method. It performs no calculations. There are no slots specifically associated with this method.

11.1.6.2 Input Change in Storage

This method allows the user to in put the change is storage as a function of time. In this way, the user controls the fill and drain dates as well as how quickly the canal fills or drains.

Slots Specific to this Method

☞ **Change in Storage Per Timestep**

Type: Periodic Slot
Units: VOLUME
Description: The user specified change in storage that should take place per timestep.
Information: This slot is used in the Outflow calculation and the Storage calculation. The user specifies when the canal starts filling/draining and the amount by which the canal is filled/drained per timestep. A positive value means the canal is filling and a negative value means the canal is draining.
I/O: Required input
Links: Not linkable

☞ **Calculated Change In Storage**

Type: Series Slot
Units: VOLUME
Description: The value compute for change in storage from the Change In Storage Per Timestep periodic slot.
Information: Usually this slot just shows the change in storage value retrieved/calculated from the periodic slot. However, if a value is input on this slot, it will be used to compute Storage instead of the Change in Storage Per Timestep slot.
I/O: Output; optional input
Links: Not linkable

☞ **Storage**

Type: Series
Units: VOLUME
Description: The amount of storage in the canal
Information: This slot is solved for based on the Change in Storage Per Timestep
I/O: Output

Links: Not linkable

When the Input Change in Storage method is selected, the Storage slot is computed and the Outflow calculation changes:

$$\text{Outflow} = \text{Available Flow} - \text{Delivered Flow} + \text{Return Flow} - \text{Change in Storage as Flow Rate}$$

$$\text{Storage} = \text{Storage}(-1) + \text{Change in Storage Per Timestep}$$

The Change in Storage is also considered when routing requests upstream.

NOTE: When routing requests upstream, the Routed Delivery Request is always zero for timesteps where the canal is draining. In other words, if the canal is draining, there can be no downstream or delivery requests routed upstream.

NOTE: If a value is input on the Calculated Change In Storage slot, it is used instead of the value determined from the Change in Storage Per Timestep periodic slot.

11.1.7 Evaporation from Storage

The Evaporation from Storage category is only available if a method is selected in the Canal Storage category. While this method is used to model evaporation from storage, the storage calculation is not affected by evaporation. Instead, evaporation is accounted for in the Outflow calculation.

11.1.7.1 None

This is the default method. It performs no calculations. There are no slots specifically associated with this method.

11.1.7.2 Input Evaporation

Slots Specific to this Method

☞ **Evaporation from Storage**

Type: Periodic Slot
Units: FLOW
Description: The volume of evaporation per unit time
Information: This periodic slot should be set up so that there is a non-zero (positive) value only on the dates where there will be a non-zero (positive) storage.
I/O: Required input
Links: Not linkable

When the Input Evaporation method is selected, the Outflow calculation changes as follows:

$$\text{Outflow} = \text{Available Flow} - \text{Delivered Flow} + \text{Return Flow} - \text{Evaporation from Storage}$$

If there is a non-zero (positive) value for Evaporation from Storage when the Storage is zero, a warning is posted.

The Evaporation from Storage is also considered when routing requests upstream.

NOTE: Neither Evaporation from Storage (nor any other gain/loss components) affect the Storage directly. The Storage slot changes only when there is a non-zero value in the Change in Storage Per Timestep slot. All gain/loss terms are handled in the Outflow calculation.

11.2 Dispatch Methods

Used to solve for Outflow and Available Flow given Inflow.

11.2.1 solveMB_givenInflow

Required Knowns

☞ **Inflow**

Required Unknowns

☞ **None**

The Canal Max Capacity and Seepage user methods are applied to the Inflow. Then Available Flow is calculated from the Flow Routing user method. Outflow is calculated as:

$$Outflow = AvailableFlow - DeliveredFlow + ReturnFlow$$

If a method is selected in the Canal Storage category, the change in storage is also removed from in the Outflow calculation:

$$\text{Outflow} = \text{Available Flow} - \text{Delivered Flow} + \text{Return Flow} - \text{Change in Storage as Flow Rate}$$

If a method is selected in the Evaporation from Storage category, the evaporation is also removed from the Outflow:

$$\text{Outflow} = \text{Available Flow} - \text{Delivered Flow} + \text{Return Flow} - \text{Evaporation from Storage}$$

Note: Time Lag Routing sets Available Flow at some future timestep. Therefore, the dispatch method may not solve for Outflow at the current timestep unless Available Flow has been set at the current timestep from dispatching at a previous timestep.

12. Diversion

A Diversion Object diverts water from either a Reach or a Reservoir. The diverted water flows through the Diversion Object and into either an AggDistributionCanal, a WaterUser, a Reach, or a Reservoir. If either a WaterUser or an AggDistributionCanal is used, the values in the Diversion (or Delivery) Requested slot should be linked to the Diversion Request slot on the Diversion Object. If either a Reach or Reservoir is linked to the downstream end of the Diversion Object, the Diversion Request slot must be input on the Diversion Object itself.

General Slots

(slots which always appear for this object)

☞ AVAILABLE FOR DIVERSION

Type: SeriesSlot

Units: FLOW

Description: amount of flow that can be diverted from an upstream Reach or Reservoir

Information:

I/O: Optional; can be calculated or set by propagation across a link

Links: May be linked to the Available For Diversion slot on a reach or reservoir.

☞ DIVERSION

Type: SeriesSlot

Units: FLOW

Description: amount diverted from the Reach or Reservoir

Information:

I/O: Output only

Links: Linked to the Diversion slot on a reach or reservoir.

☞ DIVERSION REQUEST

Type: SeriesSlot

Units: FLOW

Description: diversion requested from an upstream Reach or Reservoir

Information: Can be set as an input, particularly if there is a Reach or Reservoir linked to the downstream end of the Diversion Object. If there is an AggDistributionCanal or WaterUser linked to the downstream end, the Diversion Request slot on the Diversion Object should be linked to either the Total Delivery Request slot on the AggDistributionCanal or the Diversion Request slot on the WaterUser.

I/O: Optional; may be input, calculated or set by propagation across a link.

Links: May be linked to the Total Delivery Requested slot on an AggDistributionCanal, Total Diversion Requested on an AggDiversion Site, or Diversion Requested on a WaterUser.

☞ Diversion Shortage

Type: Series Slot

Units: FLOW

Description: The amount that the Diversion Request is shorted

Information: Computed as the Diversion Request minus the Diversion

I/O: Output only

Links: Not linkable

12.1 User Methods

12.1.1 Diversion Object Solution Direction

The methods in this category are used to add slots and control the available dispatch methods. They perform no calculations themselves.

12.1.1.1 Solve for Outflow

If this method is selected, the Diversion object will solve for its Outflow given Diversion or Diversion Requested.

SLOTS SPECIFIC TO THIS METHOD

☞ **OUTFLOW**

Type: Series Slot

Units: FLOW

Description: outflow from the Diversion Object

Information:

I/O: Output only

Links: Can be linked to the Inflow slot of a downstream object.

12.1.1.2 Solve Given Outflow

If this method is selected, the Diversion Object will solve for Diversion when Outflow is known. In this configuration, the Diversion Object can have multiple outflows using links to the Multi Outflow slot.

SLOTS SPECIFIC TO THIS METHOD

☞ **MULTI OUTFLOW**

Type: Multi Slot

Units: FLOW

Description: outflow from the Diversion Object

I/O: Can be input or set via link

Links: Can be linked to the Supply From Reservoir slot on a Water User object, or to many links on other objects.

12.1.2 Available Flow

This category contains the calculations necessary to set Available for Diversion based on either available flow in the Reach/Reservoir or on head/elevation data.

12.1.2.1 None

This is the default user method for the Available Flow category. It performs no calculations. If it is selected an error will be posted when the model is run. The user must select either Available Flow Diversion, Gravity Diversion, or Pumped Diversion for the model to run.

12.1.2.2 Available Flow Diversion

Available for Diversion is set through a link to the Available for Diversion slot on the Reach or Reservoir. The user can also input Available for Diversion if needed.

SLOTS SPECIFIC TO THIS METHOD

☞ **MAX DIVERSION**

Type: TableSlot
Units: FLOW
Description: The maximum amount of water that can pass through the Diversion Object
Information:
I/O: Optional
Links: Not linkable

☞ **MIN DIVERSION**

Type: TableSlot
Units: FLOW
Description: The lower limit on the amount of water that can be requested.
Information: Can be used if the Diversion Requests are very small and a particular flow rate is required for water to pass through the Diversion Object.
I/O: Optional
Links: Not linkable

The values in Available for Diversion either propagate across a link from the Reach or Reservoir or are input by the user. Therefore, no calculations are performed in this method.

12.1.2.3 Gravity Diversion

The Gravity Diversion method calculates Available for Diversion based on the Diversion Intake Elevation, Diversion Base Elevation, and Gravity Head Flow Table.

SLOTS SPECIFIC TO THIS METHOD

☞ DIVERSION BASE ELEVATION

Type: TableSlot

Units: LENGTH

Description: elevation of the bottom of the intake of the Diversion Object

Information:

I/O: Required input

Links: Not linkable

☞ DIVERSION INTAKE ELEVATION

Type: SeriesSlot

Units: LENGTH

Description: water surface elevation at the intake of the Diversion Object

Information:

I/O: Required

Links: Must be linked to either the Previous Pool Elevation on a Reservoir or the Water Elevation at Diversion slot on the Reach.

☞ GRAVITY HEAD FLOW TABLE

Type: TableSlot

Units: LENGTH VS. FLOW

Description: defines the relationship between Diversion Head and Max Flow into the Diversion Object

Information: Values must be input to define the relationship between Effective Diversion Head and Flow into the Diversion Object if the Gravity Diversion method is selected.

I/O: Required input

Links: Not linkable

The calculations used to determine Available for Diversion are presented below:

$$\text{Diversion Head} = \text{Diversion Intake Elevation} - \text{Diversion Base Elevation}$$

The value of Diversion Head is then used in an interpolation method to find the corresponding flow rate from the Gravity Head Flow Table. Available for Diversion is then set to the value returned from the interpolation method.

12.1.2.4 Pumped Diversion

The Pumped Diversion method calculates Available for Diversion based on the Diversion Intake Elevation, Diversion Base Elevation, and Pumped Head Flow Table.

SLOTS SPECIFIC TO THIS METHOD

☞ DIVERSION BASE ELEVATION

Type: TableSlot

Units: LENGTH

Description: elevation of the bottom of the intake of the Diversion Object

Information:

I/O: Required input

Links: Not linkable

☞ DIVERSION INTAKE ELEVATION

Type: SeriesSlot

Units: LENGTH

Description: water surface elevation at the intake of the Diversion Object

Information:

I/O: Required

Links: Must be linked to either the Previous Pool Elevation on a Reservoir or the Water Elevation at Diversion slot on the Reach.

☞ PUMPED HEAD FLOW TABLE

Type: TableSlot

Units: LENGTH VS. FLOW

Description: defines the relationship between Diversion Head and Max Flow into the Diversion Object

Information:

I/O: Required input

Links: Not linkable

The calculations used to determine Available for Diversion are presented below:

$$\text{Diversion Head} = \text{Diversion Intake Elevation} - \text{Diversion Base Elevation}$$

The value of Diversion Head is then used in an interpolation method to find the corresponding flow rate from the Pumped Head Flow Table. Available for Diversion is then set to the value returned from the interpolation method.

12.1.3 Diversion Request

Used to specify how Diversion Request is calculated.

12.1.3.1 Input Diversion Request

This method is used if the user wishes to input the Diversion Request. No calculations are performed in this method and there are no slots specifically associated with this method.

Slots Specific to this Method

☞ **None**

12.1.3.2 Percent of Available

Used to calculate Diversion Request as a percentage of the available water.

Slots Specific to this Method

☞ **Percent of Available to Divert**

Type: SeriesSlot
Units: FRACTION
Description: the percentage of the available water to be diverted
Information: Must be between 0% and 100%
I/O: Required input
Links: Not linkable

$$\text{Diversion Requested} = \text{Percent of Available to Divert} \times \text{Available For Diversion}$$

12.1.3.3 Periodic Diversion Request

This method is used to specify the Diversion Request as a function of date/season using a periodic slot.

Slots Specific To This Method

☞ **Periodic Diversion Request**

Type: Periodic Slot
Units: FLOW
Description: The diversion request as a function of date/season

Information: The Diversion Request is computed at the beginning of each timestep using the current date to look up the value from this slot.

I/O: Required input

Links: Not linkable

The Periodic Diversion Request method executes at the beginning of the run. Each timestep in the run is used to look up a value in the Periodic Diversion Request slot. The computed value is set in the Diversion Request slot.

12.1.3.4 Reservoir Level Lookup

The Reservoir Level Lookup method is used to compute the Diversion Request as a function of the date/season and the operating level of a specified reservoir.

Slots Specific To This Method

☞ Level vs Diversion Request

Type: Periodic Slot

Units: FLOW

Description: The diversion request as a function of reservoir level and date/season

Information: Each column corresponds to a reservoir's operating level. For each operating level, the diversion request is specified for each date range in the periodic slot.

I/O: Required input

Links: Not linkable

☞ Diversion Request Reservoir

Type: List Slot

Units: NONE

Description: The reservoir used to compute the diversion request

Information: The previous timestep operating level of this reservoir is used to look up the diversion request in the Level vs Diversion Request slot.

I/O: Required input

Links: Not linkable

☞ Forecast Period

Type: Scalar

Units: NONE

Description: The forecast period is a number of timesteps, including the current simulation timestep, that is used in the algorithms for calculating forecasted hydrology, regulation discharge and flood releases. This can be propagated from a computational subbasin of which the control point is a member.

Information: This slot must be input and greater than or equal to 1. If a Forecast Period is not necessary and you wish to only consider the current timestep, input a value of 1.

Information: Required Input

Links: Not Linkable

Note: If the specified reservoir is disabled and is set to Pass Inflows, no request lookup is possible. See the note **HERE (USACE_SWD.pdf, Section 4.3.1)** on the behavior in this situation.

The Reservoir Level Lookup method will execute at the beginning of each timestep. For each timestep (t) in the forecast period, the Diversion Request Reservoir's previous timestep's (current timestep - 1) operating level and the forecast timestep's date (t) will be used tolook up the diversion request value in the Level vs Diversion Request slot. This value is set on the Diversion Request slot. The Diversion is limited by the Maximum Diversion value. Because the method uses the same operating level (from the previous timestep) for all timesteps in the forecast period, it is only an approximation but allows the object to solve throughout the forecast period.

12.1.4 Diversion Pump Energy

The Diversion Pump Energy category is used to determine the Energy and Total Energy Cost based on the Diversion Intake Elevation, Outflow Elevation, Diversion, Pump Efficiency, and cost per Unit Energy. This method category is only available if Pumped Diversion is selected in the Available Flow category.

12.1.4.1 None

This is the default method. It performs no energy calculations.

12.1.4.2 Energy Equation and Cost

The Energy Equation and Cost method calculates Energy Consumption and Total Energy Cost. This method is only available when Pumped Diversion is selected in the Available Flow category.

Slots Specific to this Method

☞ **Cost per Unit Energy**

Type: TableSlot

Units: $PER ENERGY

Description: cost of each unit (MWH) of energy generated or consumed by the Diversion Object

Information: Used to determine the monetary value of the energy consumed by the Diversion Object.

I/O: Required input

Links: Not linkable

☞ **Energy Consumption**

Type: SeriesSlot

Units: ENERGY

Description: energy consumed by the Diversion Object

Information:

I/O: Output only

Links: Not linkable

☞ **Outflow Elevation**

Type: TableSlot

Units: LENGTH

Description: water surface elevation at the outflow of the Diversion Object

Information:

I/O: Required input

Links: Not linkable

☞ Pump Efficiency

Type: TableSlot

Units: DECIMAL

Description: efficiency of the pump in the Diversion Object

Information: Used in the calculation of the amount of energy consumed by the Diversion Object.

I/O: Required input

Links: Not linkable

☞ Total Energy Cost

Type: SeriesSlot

Units: $

Description: the monetary value of the energy consumed by the Diversion Object

Information:

I/O:

Links:

The engineering algorithm used in this method is described below.

$$\text{Operating Head} = \text{Diversion Intake Elevation} - \text{Outflow Elevation}$$

$$\text{Energy Consumed} = \frac{-1 \times \text{Diversion} \times \text{Density of Water} \times \text{Gravity} \times \text{Operating Head} \times \text{Timestep in Seconds}}{1000000 \times \text{Pump Efficiency}}$$

$$\text{Total Energy Cost} = \text{Energy Consumed} \times \text{Cost per Unit Energy}$$

12.2 Dispatch Methods

Used to solve for Diversion and call the user method functions. The active dispatch method(s) are dependent upon the user method selected in the Diversion Object Solution Direction category, the Available Flow category, and the Diversion Request category.

12.2.1 solveMB_givenDivHead

Solves for Diversion when Gravity Diversion or Pumped Diversion is selected in the Available Flow category and Input Diversion Request is selected in the Diversion Request category. The Solve for Outflow method must also be selected.

Required Knowns

- ☞ **Diversion Request**

- ☞ **Diversion Intake Elevation**

Required Unknowns

- ☞ **None**

If the Diversion Intake Elevation is less than or equal to Diversion Base Elevation, Available For Diversion, Diversion, and Outflow are set to zero. In this case, the water surface is at or below the Diversion Base Elevation so it is impossible for water to enter the Diversion Object.

If the Diversion Intake Elevation is greater than the Diversion Base Elevation, Available for Diversion is calculated based on the method selected in the Available Flow category. Then, if Available for Diversion is less than Diversion Request, Diversion is set equal to Available for Diversion. Otherwise, Diversion is set equal to Diversion Request. Outflow is then set equal to Diversion. Finally, the Diversion Pump Energy method, if selected, is executed.

12.2.2 solveMB_givenAvailForDiv

Solves for Diversion when Available Flow Diversion is selected in the Available Flow category and Input Diversion Request is selected in the Diversion Request category. The Solve for Outflow method must also be selected.

Required Knowns

☞ **Available For Diversion**

☞ **Diversion Request**

Required Unknowns

☞ **None**

First, the user method selected in the Available Flow category is executed. Then a temporary variable, tempRequest, is assigned the value of Diversion Request or, if Diversion Request is less than the Min Diversion Request, it is assigned the value of Min Diversion. Diversion is then set as the minimum value of tempRequest, Available For Diversion, and Max Diversion.

Outflow is set equal to Diversion.

12.2.3 solveMB_givenDivHeadPercent

Solves for Diversion when Gravity Diversion or Pumped Diversion is selected in the Available Flow category and Percent of Available is selected in the Diversion Request category. The Solve for Outflow method must also be selected.

Required Knowns

☞ **Percent of Available to Divert**

☞ **Diversion Intake Elevation**

Required Unknowns

☞ **Diversion Request**

If the Diversion Intake Elevation is less than or equal to Diversion Base Elevation, Available For Diversion, Diversion, and Outflow are set to zero. In this case, the water surface is at or below the Diversion Base Elevation so it is impossible for water to enter the Diversion Object.

If the Diversion Intake Elevation is greater than the Diversion Base Elevation, Available for Diversion is calculated based on the method selected in the Available Flow category. Then, if Available for Diversion is less than Diversion Request, Diversion is set equal to Available for Diversion. Otherwise, Diversion is set equal to Diversion Request. Outflow is then set equal to Diversion. Finally, the Diversion Pump Energy method, if selected, is executed.

12.2.4 solveMB_givenAvailablePercent

Solves for Diversion when Available Flow Diversion is selected in the Available Flow category and Percent of Available is selected in the Diversion Request category. The Solve for Outflow method must also be selected.

Required Knowns

☞ **Available For Diversion**

☞ **Percent of Available to Divert**

Required Unknowns

☞ **Diversion Request**

First, the user method selected in the Available Flow category is executed. Then Diversion Request is calculated according to the selected Diversion Request method. Then a temporary variable, tempRequest, is assigned the value of Diversion Request or, if Diversion Request is less than the Min Diversion Request, it is assigned the value of Min Diversion. Diversion is then set as the minimum value of tempRequest, Available For Diversion, and Max Diversion.

Outflow is set equal to Diversion.

12.2.5 solveMB_givenOutflow

This method solves for Diversion and Diversion Request when Outflow is known. This dispatch method is only available when the user has selected the Solve Given Outflow method in the Diversion Object Solution Direction category. Also, the Input Diversion Request method must be selected in the Diversion Request category and the Available Flow Diversion method must be selected in the Available Flow category.

Required Knowns

☞ **Multi Outflow**

Required Unknowns

☞ **Diversion**

☞ **Diversion Request**

This dispatch method sets Diversion and Diversion Request equal to the known Multi Outflow value. The Diversion is also constrained by the Min Diversion and Max Diversion values.

13. Generator

A generator represents a single generating unit within the overall Power Plant Diversion (described HERE (Section 20)). A generator has an individual capacity, efficiency and rating. The purpose of each generator is to capture the characteristics of that generator.

General Slots

(slots which always appear for this object)

☞ Diversion Requested

Type: Series
Units: Flow
Description: The computed withdrawal request for this generator.
Information:
I/O: Typically output, but could be input.
Links: Linkable

☞ Depletion Requested

Type: Series
Units: Flow
Description: The computed consumption request for this generator.
Information:
I/O: Typically output, but could be input.
Links: Linkable

13.1 User Methods

13.1.1 Diversion and Consumption Request

This category is used to specify how the Diversion and Depletion Requested will be computed.

13.1.1.1 None

No diversion or depletion requested will be computed. You can input the requests as needed.

13.1.1.2 Power Generator Requests

The Power Generator Requests method is executed at the beginning of the run (on each timestep in the run) to compute the diversion and depletion requests based on temperature lookups to get the capacity and request slot values.

Slots Specific to this Method

☞ **GENERATOR OPERATING FACTORS**

Type: Table
Units: TEMPERATURE, TEMPERATURE, FRACTION, FRACTION
Description: A table relating Dry Bulb Temperature (C), Wet Bulb Temperature (C) versus Capacity Percentage (%) and Efficiency (%).
Information: This is a 3D table (with two Y columns) of the form:

Dry Bulb Temperature (C)	Wet Bulb Temperature (C)	Capacity Fraction (%)	Efficiency (%)
20	20	70	60
20	30	65	50
20	40	60	50
25	20	70	59
25	30	64	50
25	40	59	50
30	20	69	51
30	30	63	49
30	40	57	48

I/O: Required Input
Links: Not Linkable

☞ Generator Maximum Rates

Type: Table
Units: Temperature, Vol/Energy, Vol/Energy
Description: A table relating Dry Bulb Temperature (C), Max Withdrawal Rate (Vol/Energy) and Max Consumption Rate (Vol/Energy)
Information: This is a two dimensional table with two Y columns.
I/O: Required input
Links: Not Linkable

☞ Name Plate Rating

Type: Scalar
Units: Power
Description: The maximum power that can be produced.
Information:
I/O: Required Input
Links: Not Linkable

☞ Wet Bulb Temperature

Type: Series
Units: Temperature
Description: The average wet bulb temperature (Celsius) for the timestep.
Information:
I/O: Specified as input or via a rule
Links: Not Linkable

☞ Dry Bulb Temperature

Type: Series
Units: Temperature
Description: The average dry bulb temperature (Celsius) for the timestep.
Information:
I/O: Specified as input or via a rule
Links: Not Linkable

☞ Specified Withdrawal Request

Type: Series
Units: Volume per Energy
Description: The specified withdrawal request. This value, if specified, overrides the computed max withdrawal request.
Information:
I/O: Optional input or specified via rule. Can remain unspecified.
Links: Not Linkable

☞ Specified Consumption Request

Type: Series
Units: Volume per Energy
Description: The specified consumption request. This value, if specified, overrides the computed max consumption request.
Information:
I/O: Optional input or specified via rule. Can remain unspecified.
Links: Not Linkable

☞ Capacity Fraction

Type: Series
Units: Fraction
Description: The computed fraction of the capacity.
Information:
I/O: Output Only
Links: Not Linkable

☞ Efficiency

Type: Series
Units: Fraction
Description: The computed power efficiency
Information: Note, this value is presented for information only. It is not used in any calculations.
I/O: Output only
Links: Not Linkable

☞ Adjusted Capacity

Type: Series
Units: Power
Description: The maximum power that can be produced by this generator, after adjusting for capacity fraction.
Information:
I/O: Output only
Links: Not Linkable

☞ Max Withdrawal Request

Type: Series
Units: Volume per Energy
Description: The maximum volume per energy for this generator.
Information:
I/O: Output Only
Links: Not Linkable

☞ Max Consumption Request

Type: Series
Units: Volume per Energy
Description: The maximum consumption rate for this generator.
Information:
I/O: Output Only
Links: Not Linkable

This method is executed at the beginning of the run for all timestep in the run period. The required computations at each timestep include:

Look up the specified **Wet Bulb Temperature** and **Dry Bulb Temperature** on the **Generator Operating Factors** table to get the **Capacity Fraction** and **Efficiency**. Compute the **Adjusted Capacity**:

Adjusted Capacity = **Name Plate Capacity** * **Capacity Fraction**

Look up the specified **Dry Bulb Temperature** on the **Generator Maximum Rates** table to get **Max Withdrawal Request** and **Max Consumption Request**.

Next, compute the **Diversion Requested**. If the **Specified Withdrawal Request** is valid,

Diversion Requested = **Adjusted Capacity** * **Specified Withdrawal Request** * timeConversion

Else

Diversion Requested = **Adjusted Capacity** * **Max Withdrawal Request** * timeConversion

Similarly, compute the **Depletion Requested**. If the **Specified Consumption Request** is valid,

Depletion Requested = **Adjusted Capacity** * **Specified Consumption Request** * timeConversion

Else

Depletion Requested = **Adjusted Capacity** * **Max Consumption Request** * timeConversion

The time conversion is necessary because the internal units for the requests is m3/MWH while the internal volume for flow is cms. Thus the timeConversion = 1hr/3600s.

13.2 Dispatch Methods

13.2.1 No Methods

There are no dispatch methods on the Generator.

14. Groundwater Storage

The Groundwater Storage Object models a simple fill and spill underground body of water. The Inflow to the object is either given through user input or a link from a slot on another object. Storage and Outflow are then calculated based on a mass balance performed on the object and the selected user methods. The user must input a value for Storage at the initial timestep of the run for the object to solve successfully.

The user must select a **Groundwater Outflow** method for this object; otherwise, a **RiverWare**™ error will be posted during initialization.

General Slots

SLOTS WHICH ALWAYS APPEAR FOR THIS OBJECT

☞ **STORAGE**

Type: SeriesSlot
Units: VOLUME
Description: volume of water in the object
Information: Computed by mass balance.
I/O: Output; the initial value must be input.
Links: Not linkable

14.1 User Methods

The slots described under each user method are those that are only applicable to that particular user method. They are in addition to any general slots on the object.

14.1.1 Solution Type

14.1.1.1 Single Computed Outflow

This is the default method in the Solution Type category. It will allow the groundwater object to solve with the following slots, user methods, and dispatch methods.

Slots Specific to this Method

☞ **Inflow**

Type: MultiSlot
Units: FLOW
Description: flow rate at entrance of the object
Information: May be set as input by the user or linked to the return flow of one or more other objects.
I/O: Optional; either input or set by propagation across a link
Links: Can be linked to the Return Flow of any object, Seepage on a Reach or Distribution Canal, or GW Return Flow on a WaterUser.

☞ **Outflow**

Type: SeriesSlot
Units: FLOW
Description: flow rate at exit of the object
Information:
I/O: Output only
Links: Can be linked to the Inflow of any object or to the Local Inflow slot on a Reach.

☞ **Inflow from Groundwater**

Type: MultiSlot
Units: FLOW
Description: flow into the groundwater storage object from nearby groundwater storage objects
Information: The subslots of this multislot contain the individual inflow values. A subslot is created when the users links this slot to an origin object.
I/O: Output only

Links: Linkable, usually to the Response Components slots on other Groundwater Storage objects.

14.1.1.2 Head Based Groundwater Grid

The Head Based Groundwater Grid method is used to instantiate the appropriate slots, user method categories, and dispatch methods for solution with connections to other objects (groundwater, reaches, etc.). There are no calculations specifically associated with this user method. Rather, the method is used to specify the solution type performed by the groundwater object. Note, however, that RiverWare assumes a consistent datum throughout all related objects' elevation and stages slots.

Slots Specific to this Method

☞ **Inflow from Surface Water**

Type: Multi Slot
Units: FLOW
Description: Inflow(s) to the groundwater object from the surface water system
Information: A positive value is an inflow to the groundwater object. Since this slot is a Multi Slot, it can include inflows from multiple surface water sources. If this slot is not linked, it is set to zero at the beginning of the run. It is a required known for dispatching.
I/O: Input or linked
Links: May be linked to the Seepage slot on a reach object and/or any other surface water source

☞ **Elevation**

Type: Series Slot
Units: LENGTH
Description: The water table elevation of the groundwater object. Datum should be consistent throughout all related objects' elevation and stages slots.
Information: This value is computed as a function of change in storage, Aquifer Area, and Specific Yield
I/O: Output; the initial elevation value must be input
Links: Not linkable

☞ **Previous Water Table Elevation**

Type: Series Slot
Units: LENGTH
Description: The previous water table elevation
Information: This slot is used to control dispatching between connected groundwater objects. It will have the same value as the Elevation slot at the previous timestep. When the Elevation is computed during dispatching, the same value will be set on the Previous Water Table Elevation slot at the NEXT timestep

to trigger the dispatching of connected groundwater objects at the next timestep.
I/O: Output only
Links: Linked to the Elevation Upstream/Downstream/Left/Right Previous slots on a connected groundwater object, and/or the Previous Water Table Elevation slot on a reach.

☞ **SPECIFIC YIELD**
Type: Scalar Slot
Units: NONE
Description: The ratio of the volume of water added or removed directly from the saturated aquifer to the resulting change in the volume of aquifer below the water table
Information: The specific yield is used to compute water table elevation as a function of aquifer area and change in storage
I/O: Required input
Links: Not linkable

☞ **AQUIFER AREA**
Type: Scalar Slot
Units: AREA
Description: The horizontal area of the aquifer. The surface area of the water table.
Information: Used to compute aquifer elevation and evapotranspiration.
I/O: Input or computed (see **HERE (Section 14.1.5.2)**)
Links: Not linkable

☞ **SPECIFIED INFLOW**
Type: Series Slot with Periodic Input
Units: FLOW
Description: Inflows to the groundwater object typically from other groundwater objects that are not modeled. This slot provides a place to specify any inflows that aren't classified as other inflows.
Information: This value can be positive (into the object) or negative (out of the object). Specified Inflows that are negative can cause the storage to become negative.
I/O: Input as a periodic or series data. Specify a single value if you have a constant value. If no value is specified, it is assumed the value is zero, but that is not shown on the slot.
Links: Not Linkable

14.1.1.3 Head Based Boundary Condition

The **Head Based Boundary Condition** method is used to add the appropriate slots, user method categories, and dispatch methods for solution of a groundwater network linked to a reservoir. This method triggers the solveGWInflows_givenElevations dispatch method HERE (Section 14.2.3).

An object with this method selected should be linked to a single Reservoir object with the **Linked Seepage** method selected HERE (Objects.pdf, Section 24.1.24.5). It should also be linked to one or more Groundwater objects with the **Head Based Groundwater Grid** method selected for the Solution Type. There are no calculations specifically associated with this user method. Rather, the method is used to specify the solution type performed by the Groundwater object. Note, however, that RiverWare assumes a consistent datum throughout all related objects' elevation and stages slots.

There is no **Storage** on a Groundwater object with this method selected. It is added only as a means to convey **Seepage** water from the linked Reservoir into the linked Groundwater object. The **Elevation** on this object will be the same as the **Pool Elevation** on the Reservoir.

When this new method is selected, the following categories will be available:

- **Lateral Link Direction**
- **Deep Percolation**
- **Groundwater Conductance**

The **Groundwater Evaporation**, **Groundwater Evapotranspiration**, **Groundwater Pumping**, and **Groundwater Water Quality**, methods will not be available.

Slots Specific to this Method

☞ Elevation Previous

Type: Series
Units: LENGTH
Description: Represents the previous timestep Pool Elevation of the linked Reservoir; datum should be consistent throughout all related objects' elevation and stages slots
Information: This slot is used to control dispatching between connected groundwater objects. It will have the same value as the Elevation slot at the previous timestep. The value will come from the linked Reservoir dispatching and propagating the value across the link.
I/O: Output only
Links: Typically linked to the **Previous Elevation** slot on a Reservoir

☞ Elevation

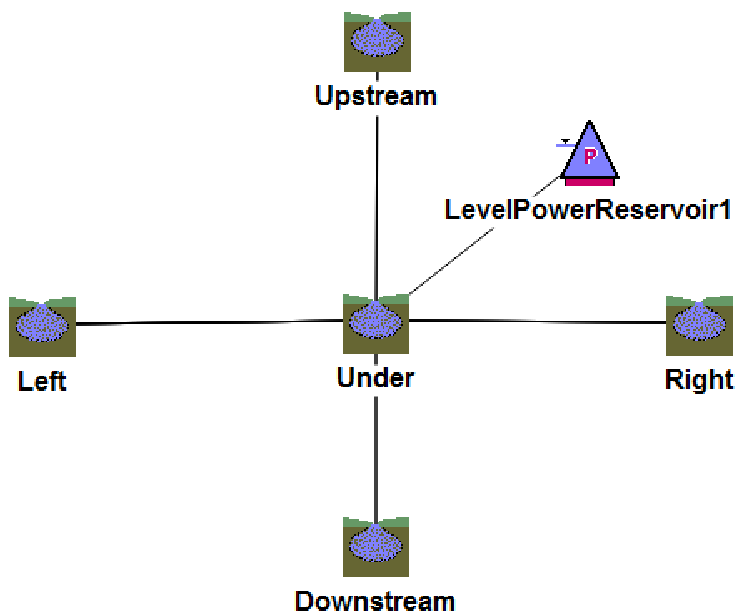
Type: Series
Units: LENGTH
Description: Represents the Pool Elevation of the linked Reservoir; datum should be consistent throughout all related objects' elevation and stages slots
Information:

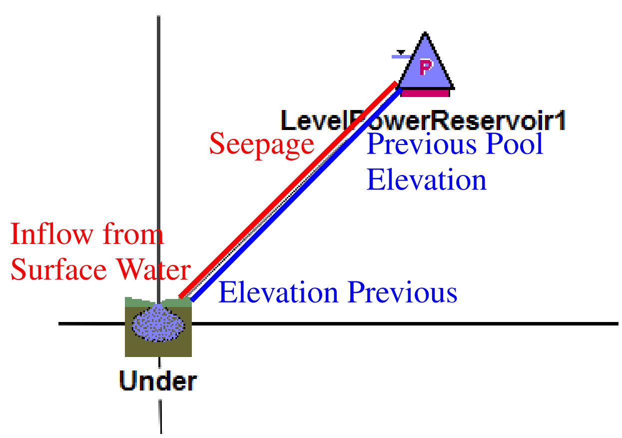
I/O: Output only
Links: Not linkable

☞ Inflow from Surface Water

Type: Multi Slot
Units: Flow
Description: Inflow(s) to the groundwater object from the surface water system, typically from a linked Reservoir
Information: A positive value is an inflow to the groundwater object.
I/O: Output only
Links: Typically linked to the **Seepage** slot on a Reservoir object; although this is a multi slot, for this method it can only be linked to a single slot. Linking this slot to multiple slots will cause an error.

The typical linking to connect a groundwater network to a Reservoir object is shown at right. The object “Under” would have the **Head Based Boundary Condition** method selected for the Solution Type and is used to convey the **Seepage** from the Reservoir to the laterally linked Groundwater objects. The laterally linked Groundwater objects would have the **Head Based Groundwater Grid** method selected for the Solution Type. There may be 1-4 laterally linked Groundwater objects. The appropriate **Elevation Previous** and **Flow Factor** slots should be linked between the “Under” Groundwater object and the laterally linked Groundwater objects. The linking with the Reservoir object is illustrated at right.





14.1.1.4 Link to MODFLOW GW

Note: RiverWare's connection with MODFLOW is currently not functional. This method has been disabled and cannot be selected. An error will be posted at model load if this method was previously selected. Contact CADSWES for help.

The "Link to MODFLOW GW" method allows the groundwater object to be linked with MODFLOW. Data transferred between RiverWare and MODFLOW, as well as any interpolation and summation of this data, will be handled within the computational subbasin framework. Note, RiverWare assumes a consistent datum throughout all related objects' elevation and stages slots.

Click **HERE (Section 7.2.1)** to view the RiverWare - MODFLOW Connection Functionality Guide. A description of the Groundwater object's specific data configuration is presented in that guide **HERE (Section 7.2.1.3.3)**.

Slots Specific to this Method

☞ Lateral Flux from MODFLOW

Type: Series Slot
Units: Flow
Description: The flux between the MODFLOW model lateral boundary cell(s) head and the RiverWare GroundWater head calculated by MODFLOW
Information: A positive value is an inflow to the groundwater object.
I/O: Output only
Links: Linkable

☞ Aquifer Area

Type: Scalar Slot
Units: Area
Description: The horizontal area of the aquifer or the surface area of the water table.
Information: Used to compute elevation. May be used as a calibration parameter.
I/O: Required input
Links: Not linkable

☞ Elevation

Type: Series Slot
Units: Length
Description: The water table elevation on the groundwater object
Information: This value is computed as a function of Percolation, GroundWater return flow, and Lateral Flux.
I/O: Output only
Links: Linkable

☞ Previous Water Table Elevation

Type: Series Slot
Units: Length
Description: The previous water table elevation on the groundwater object
Information: This slot is used to control dispatching between connected groundwater objects. It will have the same value as the Elevation slot at the previous timestep. When the Elevation is computed during dispatching, the same value will be set on the Elevation Previous slot at the NEXT timestep to trigger the dispatching of connected groundwater objects at the next timestep.
I/O: Output only
Links: Linkable

☞ Inflow from Surface Water

Type: Series Slot
Units: Flow
Description: Inflow(s) to the groundwater object from the surface water system
Information: A positive value is an inflow to the groundwater object. If this slot is not linked, it is set to zero at the beginning of the run. It is a required known for dispatching.
I/O: Optional
Links: Usually linked to the GW Return Flow slot on the Water User object

☞ Specific Yield

Type: Scalar Slot
Units: None
Description: The ratio of the volume of water added or removed directly from the saturated aquifer to the resultingchange in the volume of aquifer below the water table
Information: The specific yield is used to compute the GW head elevation as a function of aquifer area and change in storage
I/O: Required input
Links: Not linkable

14.1.2 Groundwater Outflow

This method category is dependent upon the selection of the Single Computed Outflow in the Solution Type category. The user selectable methods in this category are used to calculate Storage and Outflow from the Groundwater object based on a given Inflow.

14.1.2.1 None

This is the default user method for the **Groundwater Outflow** category. It performs no calculations. If it is selected, **RiverWare**™ gives an error because it is unable to calculate either an Outflow from or a Storage for the Groundwater Storage object.

14.1.2.2 Table Flow

The Table Flow method calculates the outflow based on the previous timestep's Storage and a table input by the user.

Slots Specific to this Method

☞ **Storage Outflow Table**

Type: TableSlot
Units: VOLUME vs. FLOW
Description: outflow for any given volume of storage
Information:
I/O: Required input
Links: Not linkable

Outflow is calculated by a linear interpolation of the Storage Outflow Table using the Storage calculated at the previous timestep.

14.1.2.3 Linear Flow

The Linear Flow method calculates Outflow as a linear function of the previous timestep's Storage value.

Slots Specific to this Method

☞ **GW Alpha Param**

Type: TableSlot
Units: PER TIME
Description: an empirical parameter used in the Outflow equation
Information:
I/O: Required input
Links: Not linkable

Outflow is calculated by an empirical equation.

$$\text{Outflow} = \text{GW Alpha Param} \times \text{Storage (t-1)}$$

14.1.2.4 Current Storage Linear Flow

The Current Storage Linear Flow method calculates Outflow as a linear function of the Average Storage at the current timestep.

Slots Specific to this Method

☞ **Average Storage**
Type: SeriesSlot
Units: VOLUME
Description: the average storage over the timestep
I/O: Output only
Links: Usually not linked

☞ **GW Alpha Param**
Type: TableSlot
Units: PER TIME
Description: an empirical parameter used in the Outflow equation
I/O: Required input
Links: Not linkable

At each timestep the method first solves for instantaneous storage, then the Average Storage, and lastly Outflow.

The three equations are:

$$\text{Storage(t)} = \frac{\text{InflowVolume} + \text{Storage(t-1)} \cdot \text{(C-alphaCoeff/2)}}{\text{C+alphaCoeff/2}}$$

$$\text{averageStorage(t)} = \frac{\text{Storage(t)} + \text{Storage(t-1)}}{2}$$

$$\text{Outflow(t)} = \text{alphaCoeff} \cdot \text{averageStorage(t)}$$

14.1.2.5 Lagged Linear Flow

The Lagged Linear Flow method calculates Outflow linearly as a function of the alpha coefficient times some previous storage value.

Slots Specific to this Method

☞ **GW Alpha Param**
Type: TableSlot

Units: PER TIME
Description: an empirical parameter used in the Outflow equation
I/O: Required input
Links: Not linkable

☞ **Lag Time**
Type: TableSlot
Units: TIME
Description: the amount of time lag between a Storage and its corresponding Outflow
I/O: Required input
Links: Not linkable

Outflow is computed using the following equation:

$$\text{Outflow(t)} = \text{alphaCoeff} \cdot \text{Storage(- lagTime)}$$

Depending on the length of the lag time, one or more storage values may need to be given prior to the required initial storage in order to compute the first outflows. If these values are not input by the user, the method assumes that storage has been constant and uses the initial storage value (which is always required user input.) This method uses instantaneous end of timestep storage values.

14.1.2.6 Exponential Flow

The Exponential Flow method calculates Outflow as an exponential function of the previous timestep's Storage value.

Note: This calculation is carried out in ***user units***. The value set in the GW Beta Param slot, as set by the user, must be consistent with the ***user units*** for the Storage slot. Changing the user units on the Storage slot will produce a different result. There will not be an automatic conversion of the GW Beta Param value.

Slots Specific to this Method

☞ **GW Alpha Param**
Type: TableSlot
Units: PER TIME
Description: an empirical parameter used in the Outflow equation
Information:
I/O: Required input
Links: Not linkable

☞ **GW BETA PARAM**

Type: TableSlot
Units: NONE
Description: an empirical parameter used in the Outflow equation
Information: This value must be consistent with the user units for the Storage slot. If the user units on the Storage slot are changed, the value in this slot must be changed manually to correspond to the new units. Otherwise different results will be produced.
I/O: Required input
Links: Not linkable

Outflow is calculated by an empirical equation. The exponential term is computed in user units (i. e. Storage(t-1) is converted to user units, raised to the power of GW Beta Param, and then converted back to the standard **RiverWare**™ units before being multiplied by GW Alpha Param):

$$\text{Outflow} = \text{GW Alpha Param} \times \text{Storage (t-1)}^{\text{GW Beta Param}}$$

14.1.2.7 Impulse Response

The Impulse Response method calculates Outflow using a set of impulse response outflow coefficients.

SLOTS SPECIFIC TO THIS METHOD

☞ **IMPULSE**

Type: SeriesSlot
Units: FLOW
Description: the impulse is the inflow minus pumped flow
I/O: Output only
Links: Usually not linked

☞ **NUMBER OF OUTFLOW COEFFS**

Type: TableSlot
Units: NONE
Description: the number of impulse response coefficients to use
Information: This single integer value is the number of values that need to be entered in the Outflow Coeffs slot.
I/O: Required input
Links: Not linkable

☞ **OUTFLOW COEFFS**

Type: TableSlot
Units: NONE

Description: the impulse response coefficients
Information: The same number of coefficients must be input as the value in the Number of Outflow Coeffs slot. The input will be in rows.
I/O: Required input
Links: Not linkable

This method first calculates the impulse to groundwater as a function of inflow and pumped flow:

$$\text{Impulse} = \text{Inflow} - \text{PumpedFlow}$$

Outflow is computed as a function of current and previous impulse values and the outflow coefficients,

$$\text{Outflow}_t = C_0\text{Impulse}_t + C_1\text{Impulse}_{t-1} + ... \\ + C_{ncoeff-1}\text{Impulse}_{t-ncoeff-1} + C_{ncoeff}\text{Impulse}_{t-ncoeff}$$

14.1.3 Excess GW Storage

This method category is dependent upon the selection of the Single Computed Outflow in the Solution Type category. The Excess GW Storage method is used to specify how excess groundwater will be modeled.

14.1.3.1 No Excess Storage

This is the default method. It performs no calculations. There are no slots specifically associated with this method.

14.1.3.2 Excess Storage to Outflow

Used if the user wants to specify a maximum possible storage in the aquifer. Any calculated storage that exceeds this value is converted to Outflow.

SLOTS SPECIFIC TO THIS METHOD

☞ **MAX GW CAPACITY**

Type: TableSlot
Units: VOLUME
Description: maximum storage capacity of the groundwater system
Information:
I/O: Optional; this slot is only used if it is input.
Links: Not linkable

The calculated Storage value is first checked against Max GW Capacity. If Storage is greater than Max GW Capacity, excess storage is calculated as the difference between Storage and Max GW Capacity. Then, Storage is set equal to Max GW Capacity. The amount of excess storage is converted to a flow rate and added to the Outflow. If Storage is less than Max GW Capacity, nothing is done.

14.1.3.3 Excess Storage Divert

Used if the user wants to specify a maximum possible storage in the aquifer. Any calculated storage that exceeds this value is converted to a flow value and set in a separate, linkable slot.

SLOTS SPECIFIC TO THIS METHOD

☞ **MAX GW CAPACITY**

Type: TableSlot
Units: VOLUME
Description: maximum storage capacity of the groundwater system
Information:

I/O: Optional; this slot is not used if it is not input.
Links: Not linkable

☞ EXCESS GW OUTFLOW

Type: SeriesSlot
Units: FLOW
Description: represents the required flow rate to dissipate any excess storage
Information: Calculated as the excess storage (Storage minus Max GW Capacity) divided by the timestep length.
I/O: Output only
Links: Can be linked to the Inflow slot on any object, the Return Flow slot on any object, or the Local Inflow slot on the Reach.

The calculated Storage value is first checked against Max GW Capacity. If Storage is greater than Max GW Capacity, excess storage is calculated as the difference between Storage and Max GW Capacity. Then, Storage is set equal to Max GW Capacity. The amount of excess storage is converted to a flow rate and the value is set on the Excess GW Outflow slot. If Storage is less than Max GW Capacity, nothing is done.

14.1.4 GW Deep Percolation

These methods are used to model vertical losses from groundwater storage to a deeper, underlying aquifer.

14.1.4.1 None

This is the default method. It performs no calculations. There are no slots specifically associated with this method.

14.1.4.2 Input Percolation

This method is used if the user want to input the amount of vertical loss or percolation.This method is dependent upon the selection of the Single Computed Outflow in the Solution Type category.

Slots Specific to this Method

☞ **Percolation**

Type: SeriesSlot
Units: FLOW
Description: the flow rate of water lost vertically to deeper groundwater storage
Information:
I/O: Optional; if not input it is set to zero.
Links: May be linked to the Inflow slot on a Groundwater Storage object.

A local variable, storage flow, is calculated as the previous timestep's Storage value divided by the length of the current timestep. This represents the flow rate required to drain the aquifer of all its stored water in the current timestep. If the input value of Percolation is greater than storage flow, an error is posted and the run is aborted. Otherwise, Percolation is converted to a volumetric value and is included in the mass balance (see Dispatch Methods).

14.1.4.3 Linear Percolation

This method calculates Percolation in the same way Outflow is calculated by the Linear Flow method in the **Groundwater Outflow** category. This method is dependent upon the selection of the Single Computed Outflow in the Solution Type category.

Slots Specific to this Method

☞ **Percolation**

Type: SeriesSlot
Units: FLOW
Description: the flow rate of water lost vertically to deeper groundwater storage
Information:

I/O: Output only
Links: May be linked to the Inflow slot on a Groundwater Storage object.

☞ **PERCOLATION ALPHA PARAM**
Type: TableSlot
Units: PER TIME
Description: an empirical parameter used in the equation to calculate Percolation
Information:
I/O: Required input
Links: Not linkable

Percolation is calculated from the following equation:

$$\text{Percolation} = \text{Storage(t-1)} \times \text{Percolation Alpha Param}$$

14.1.4.4 Exponential Percolation

This method calculates Percolation in the same way Outflow is calculated by the ExponentialFlow method in the **Groundwater Outflow** category. This method is dependent upon the selection of the Single Computed Outflow in the Solution Type Category.

Note: This calculation is carried out in ***user units***. The value set in the Percolation Beta Param slot, as set by the user, must be consistent with the ***user units*** for the Storage slot. Changing the user units on the Storage slot will produce a different result. There will not be an automatic conversion of the Percolation Beta Param value.

SLOTS SPECIFIC TO THIS METHOD

☞ **PERCOLATION**
Type: SeriesSlot
Units: FLOW
Description: the flow rate of water lost vertically to deeper groundwater storage
Information:
I/O: Output only
Links: May be linked to the Inflow slot on a Groundwater Storage object.

☞ **PERCOLATION ALPHA PARAM**
Type: TableSlot
Units: PER TIME
Description: an empirical parameter used in the equation to calculate Percolation
Information:
I/O: Required input
Links: Not linkable

☞ **PERCOLATION BETA PARAM**

Type: TableSlot
Units: NONE
Description: an empirical parameter in the equation to calculate Percolation
Information: This value must be consistent with the user units for the Storage slot. If the user units on the Storage slot are changed, the value in this slot must be changed manually to correspond to the new units. Otherwise different results will be produced.
I/O: Required Input
Links: Not Linkable

Percolation is calculated by an empirical equation. The exponential term is computed in user units (i. e. Storage(t-1) is converted to user units, raised to the power of Percolation Beta Param, and then converted back to the standard **RiverWare**™ units before being multiplied by Percolation Alpha Param):

$$\text{Percolation} = \text{Percolation Alpha Param} \times \text{Storage(t-1)}^{\text{Percolation Beta Param}}$$

14.1.4.5 Head Based Percolation

This method is dependent upon the selection of the Head Based Groundwater Grid or the Link to MODFLOW GW in the Solution Type category. This method is used to calculate vertical flow to a deeper aquifer or to incorporate boundary conditions.

☞ **PERCOLATION**

Type: Series Slot
Units: FLOW
Description: The flow to the deep aquifer
Information: Computed based on the previous timestep elevation difference. A positive value represents an outflow to the deep aquifer.
I/O: Typically output, but can be input or set by a rule.
Links: May be linked to another groundwater object

☞ **DEEP AQUIFER CONDUCTANCE**

Type: Scalar Slot
Units: AREA PER TIME
Description: The conductance between the groundwater storage object and the deep aquifer
Information:
I/O: Input or computed as specified HERE (Section 14.1.5).
Links: Not linkable

☞ **DEEP AQUIFER ELEVATION**

Type: Series Slot

Units: LENGTH
Description: The elevation/head associated with the deep aquifer. Datum should be consistent throughout all related objects' elevation and stages slots.
Information: The previous timestep (beginning of timestep) value is used to compute Percolation
I/O: Required input
Links: Not linkable

The Head Based Percolation method is called from the dispatch method. Percolation is computed by the following equation.

$$\text{Percolation} = \text{Deep Aquifer Conductance} \times (\text{Elevation Previous} - \text{Previous Deep Aquifer Elevation})$$

Note, the percolation cannot be larger than the previous Storage, converted to a flow. The computed Percolation value is passed back to the dispatch method and used in the mass balance equation to solve for the current Storage.

14.1.5 Groundwater Conductance

Methods in this category allow you to choose how you wish to specify conductance.

This category is available when any of the non-default, 15 flow methods in the **Lateral Link Direction** category is selected. It is also available when **Head Based Percolation** is selected in the **GW Deep Percolation** category. That is, it is available for any method that instantiates one of the conductance slots on the groundwater object.

Note: You can not mix use of these two approaches across linked groundwater objects. That is, groundwater objects that are calculating conductance cannot be linked to groundwater objects specifying conductance.

14.1.5.1 Specify Conductance

This is the default method and does not instantiate any new slots. You must specify each of the conductance values (Conductance Left, Conductance Right, etc...)

14.1.5.2 Compute Conductance

This method allows you to specify hydraulic conductivity and the geometry for each flow component. Then the object uses information on the object and on adjacent objects to determine the conductance in each direction. The calculation uses the following equation to take the geometric mean of conductance:

$$\text{Conductance} = \frac{2(K_i \times K_{i+1} \times \text{Area}_{\text{i to i+1}})}{K_{i+1} \times \text{Distance}_i + K_i \times \text{Distance}_{i+1}}$$

where i is the given object, and i+1 is the connected object. K is the hydraulic conductivity, $Area_{i\ to\ i+1}$ is the area shared by object i and i+1 that is perpendicular to the flow, and *Distance* is the aquifer's distance in the direction of the flow.

THE FOLLOWING SLOTS ARE ADDED:

☞ **HYDRAULIC CONDUCTIVITY**

Type: Scalar Slot
Units: VELOCITY,
Description: This slot contains the hydraulic conductivity, Kx. This represents the conductivity for lateral flows in/out of the groundwater object in the left and right direction.
Information: This value is also used to compute the conductivity in the upstream and downstream directions by dividing by the Anisotropy Ratio.
I/O:
Links: Not linkable

☞ Anisotropy Ratio

Type: Scalar Slot
Units: No Units
Description: This slot contains a ratio, used to adjust the hydraulic conductivity in the upstream/downstream or left/right direction.
Information: This ratio represents the ratio of hydraulic conductivity in Left-Right direction to the Up-Down direction (Kx/Ky). It is applied when computing conductance.
I/O: If not specified, it defaults to 1.0.
Links: Not linkable

☞ Aquifer Length

Type: Scalar
Units: Length
Description: Length of the groundwater object in the upstream/downstream direction at the centroid. This is considered the Y direction
Information:
I/O: Required input if flow can go left, right, or to deep percolation
Links: Not Linkable

☞ Aquifer Width

Type: Scalar
Units: Length
Description: Length of the groundwater object in the left/right direction at the centroid. This is considered the X direction
Information:
I/O: Required input if flow can go upstream, downstream, or to deep percolation
Links: Not Linkable

☞ Aquifer Thickness

Type: Scalar
Units: Length
Description: Length of the groundwater object in the left/right direction at the centroid. This is considered the Z direction
Information:
I/O: Required input if flow can go left, right, upstream or downstream.
Links: Not Linkable

☞ Deep Aquifer Hydraulic Conductivity

Type: Scalar
Units: Velocity
Description: Conductivity to the deep aquifer
Information:

I/O: Required Input
Links: Not Linkable

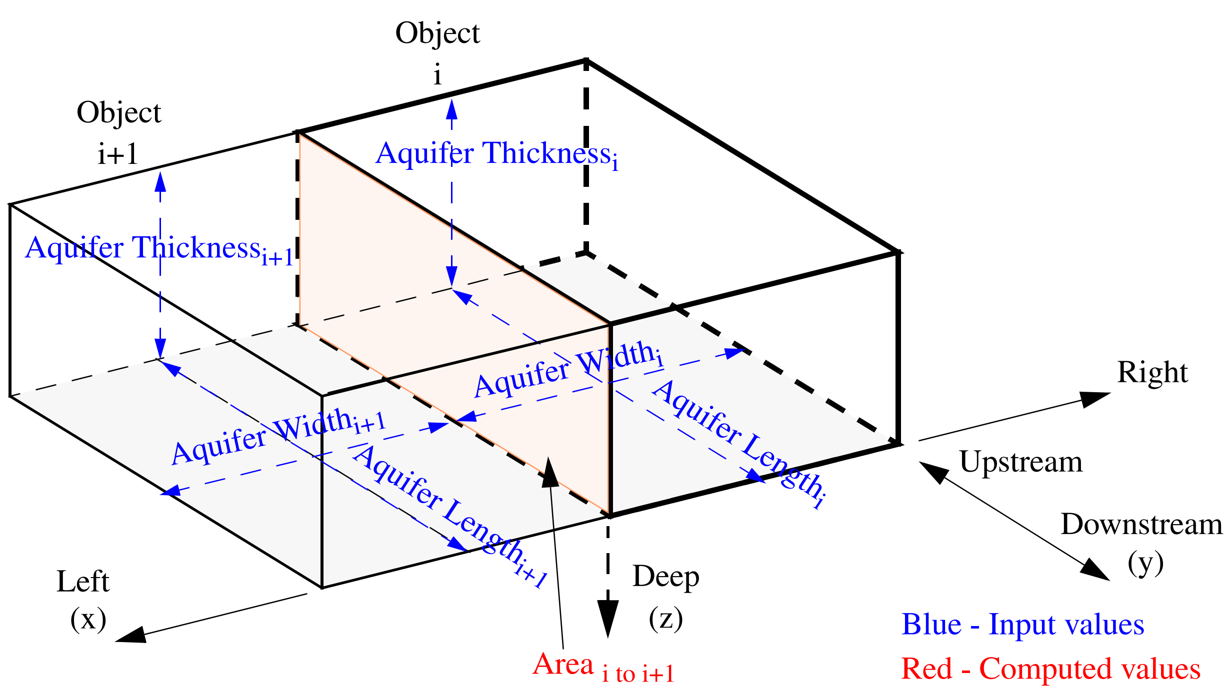
☞ DEEP AQUIFER DEPTH

Type: Scalar
Units: LENGTH
Description: Distance from the groundwater object to the deep aquifer.
Information: This value is only used to compute conductance
I/O: Required Input
Links: Not Linkable

For mass conservation across objects, the Conductance must be identical on adjacent objects, i.e. **Object$_{i+1}$.Conductance Right** must equal **Object$_i$.Conductance Left**. For the conductance to be identical, the area shared by adjacent objects must be the same.

To prevent requiring that you, the user, specify that the shared area of adjacent objects are the same, instead the computation will average the areas between adjacent cells. One way to think of this is that the length, width and thickness are measured/specified at the centroid of the object.

Specifying dimensions at the centroid allows you to input dimensions that are not the same on adjacent objects. This approach does violate the assumption that the cells are rectangular prisms, but in reality, the groundwater objects are rarely rectangular prisms. You can specify the same dimensions on adjacent objects and it will model rectangular prisms.



Because the geometry is input, there are some assumptions:

- Left/Right is perpendicular to the plane defined by Length and Thickness
- Left/Right is in the same plane as the Width.
- Upstream/Downstream is perpendicular to the plane defined by Width
- Upstream/Downstream is perpendicular to the plane as the Length.
- On adjacent objects, Left is connect to Right and vice versa.
- On adjacent objects, Upstream is connected to Downstream and vice versa.

The actual sense of the left/right/upstream/downstream does not matter as long as it is consistent.

COMPUTATION DETAILS:

The method is executed at the start of the run. If a method is selected (like Deep Percolation) and the appropriate values in the above scalars are missing, an error will be issued and the run will be stopped.

The following is performed for each groundwater object (computation of conductance in the Left/Right direction is used as an example):

1. Compute the Aquifer Area as

$$\text{Aquifer Area} = \text{Aquifer Length} \times \text{Aquifer Width}$$

Note: The Aquifer Area is registered as having the Aquifer Length as its “Source” slot. When a table slot has a source slot, it becomes read-only and displays a cross hatch over the data. It also provides a note indicating the source slot used to compute the data. The source slot is set at beginning of run, so the user must initialize the run to see the read-only status.

2. Determine the object (i+1) linked to the Elevation Upstream/Downstream/Left/Right Previous. For example, when it is computing Conductance Left, it will get the slot linked to Elevation Left Previous. From that linked slot, the method will get the parent object and then the appropriate parameters, Thickness, Length, Width, K.

3. Compute the average of the adjacent cell’s dimensions and compute the area shared by the two cells. The area shared between the two cells is the average of their dimensions.

$$\text{Area}_{\text{i to i+1}} = \left(\frac{Length_i + Length_{i+1}}{2}\right) \times \left(\frac{Thickness_i + Thickness_{i+1}}{2}\right)$$

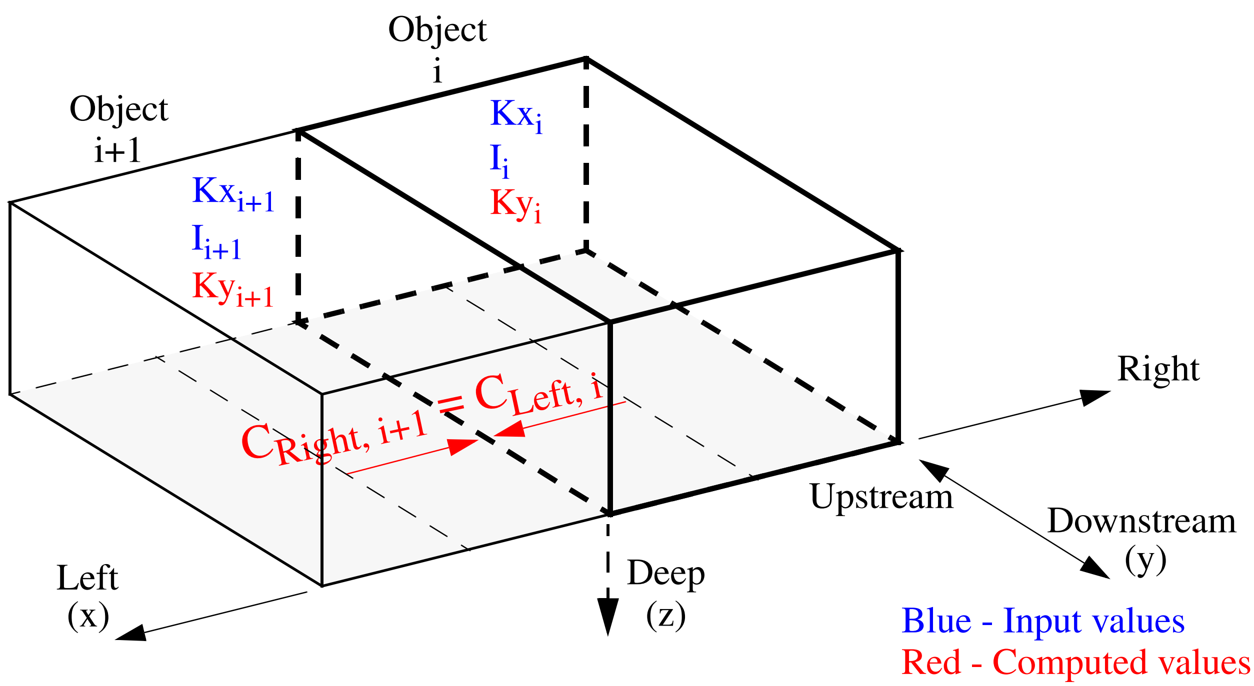
4. The Conductance (C) for each flow component is computed using the information on adjacent objects to compute the conductance values for each direction. This computation takes the harmonic mean of Hydraulic Conductivity (K) according to the following equation (Left/Right is used as an example):

$$\text{Conductance Right}_{i+1} = \frac{2(Kx_i \times Kx_{i+1} \times \text{Area}_{\text{i to i+1}})}{Kx_{i+1} \times \text{Aquifer Width}_i + Kx_i \times \text{Aquifer Width}_{i+1}} = \text{Conductance Left}_i$$ [1]

where i is the given object, and i+1 is the connected object. In the equation, Kx is:

$$Kx = K_{LeftRight} = \text{Hydraulic Conductivity}$$

The following figure shows the values input and the computed values for this computation:



5. This computation is repeated for each side: Upstream (using Ky), Downstream (Ky), and Right (Kx) using the appropriate dimensions. For the upstream and downstream directions, Ky is used:

$$K_{\text{UpDown}} = Ky = \frac{\text{Hydraulic Conductivity (Kx)}}{\text{Anisotropy Ratio}}$$

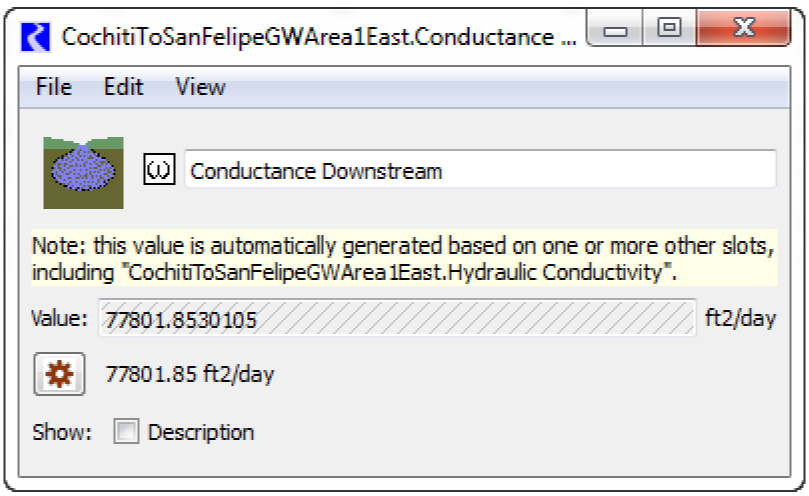
6. Next, the deep aquifer conductance will be computed:

1.See eqn 2.5.6 in Charbeneau, Randall J, *Groundwater Hydraulics and Pollutant Transport*, Prentice Hall, 2000.

$$\text{Deep Aquifer Conductance} = \text{Deep Aquifer Hydraulic Conductivity} \times \frac{\text{Aquifer Length} \times \text{Aquifer Width}}{\text{Deep Aquifer Depth}}$$

The computed conductances will be set on the appropriate scalar conductance slot. If there are values already on the slot, they will be overwritten.

Note: The appropriate conductance slot(s) will be registered as having a “Source” slot. When a table slot has a source slot, it becomes read-only and displays a cross hatch over the data. See screenshot below. It also provides a note indicating the source slot used to compute the data. The source slot is set at beginning of run, so the user must Initialize the run to see the read-only status. If you deselect this method, you must initialize the run again to un-set the source slot.



14.1.6 Groundwater Pumping

These methods are used to model pumping from a groundwater aquifer. The solution strategy is based on mass balance equations.

14.1.6.1 None

This is the default method. It performs no calculations. There are no slots specifically associated with this method.

14.1.6.2 Input Pumping

This method is designed to be used with the conjunctive use methods on the Agg Diversion Site and WaterUser objects. It models groundwater pumping by simply removing a given amount of water from storage. This method is dependent upon the selection of the Single Computed Outflow in the Solution Type category.

Slots Specific to this Method

☞ **Available For Pumping**

Type: SeriesSlot
Units: FLOW
Description: the amount of water that can be pumped from storage
Information: If not input, it is calculated based on the previous timestep's storage value and the current timestep's outflow and percolation values. This slot is usually linked to the Available Supplemental Water slot on the Agg Diversion Site
I/O: Optional
Links: May be linked to the Available Supplemental Water slot in the Agg Diversion Site object.

☞ **Pumped Flow**

Type: Multi Slot
Units: FLOW
Description: the amount of water removed (per timestep) from groundwater storage
Information: Can be input by the user but is usually linked to the Supplemental Diversion slot on the Agg Diversion Site or WaterUser.
I/O: Optional
Links: May be linked to the Supplemental Diversion slot on the Agg Diversion Site or Water User.

☞ **Water Table Elevation**

Type: SeriesSlot
Units: LENGTH
Description: elevation of the groundwater table

Information: This slot is only used to hold data for calibration purposes. It is not used in any calculations and is never calculated.
I/O: Optional
Links: Not linkable

If Available For Pumping is not input by the user, it is calculated from the previous timestep's storage and the current timestep's outflow and percolation. A local variable, storage flow, is calculated as the previous timestep's Storage value divided by the length of the current timestep. This represents the flow rate required to drain the aquifer of all its stored water in the current timestep. Available For Pumping is then calculated as:

$$\text{Available For Pumping} = Max(\text{storage flow} - \text{Outflow} - \text{Percolation}, 0.0cms)$$

If a GW Deep Percolation method is not selected, Percolation is assumed to be zero.

Pumped Flow should receive its value from a link to Supplemental Diversion on the Agg Diversion Site or Water User. It is then converted to a volumetric value, volume pumped (Pumped Flow multiplied by the timestep length), and used in the mass balance equation (see Dispatch Methods).

14.1.6.3 Input Pumped Flow

This method is designed to be used with the conjunctive use methods on the Agg Diversion Site and WaterUser objects. It models groundwater pumping by simply removing a given amount of water from storage. Flow may be input or linked. When linked the water user(s) will determine the pumped flow based on the conjunctive use and supplemental diversion calculations. This method is dependent upon the selection of the Head Based Groundwater Grid in the Solution Type category

☞ Pumped Flow

Type: Multi Slot
Units: FLOW
Description: Water removed from the aquifer by pumping (expressed as a flow rate)
Information: Removed from storage via the mass balance equation
I/O: Input or linked
Links: Usually linked to the Supplemental Diversion slot on a water user object

☞ Available For Pumping

Type: Series Slot
Units: FLOW
Description: The water available for pumping on a given timestep
Information: If not input, the Available For Pumping is the current storage (before pumping is removed).
I/O: Optional; computed if not input

Links: Usually linked to the Available Supplemental Water on a water user object.

This method is called from the SolveGWMB_GivenPreviousElevations dispatch method.

If Available For Pumping (at t+1) is not input by the user, it will be set as the current, computed storage value converted to a flow rate minus percolation minus ET as a flow. This value will propagate to the linked water user objects which can then dispatch and compute their supplemental diversions. These values return to the groundwater object in the Pumped Flow slot. The groundwater object will then re-dispatch and recompute storage given the new values for Pumped Flow. The full equation is:

$$\text{Available for Pumping(t+1)} = Max(\text{Storage as flow}(t) - \text{percolation (t+1)} - \text{ET Volume as flow}(t+1), 0.0cms)$$

Variables in the above equation that are not defined (due to the method selection) or are unknown are assumed to be zero. The Available for Pumping value will propagate to the linked water user objects which can then dispatch and compute their Diversion which is passed back to the Groundwater object as Pumped Flow. The current storage will be computed by the mass balance equation described in the SolveGWMB_GivenPreviousElevations dispatch method.

NOTE: Since the same Available For Pumping value can be referenced by several water users, the water users could pump the aquifer dry if the storage is small and each water user pumps up to the Available for Pumping. There is no mechanism to short water users or prioritize pumping if the demand exceeds the Available For Pumping. Rather, each water user can take up to the available and the aquifer storage will go negative. If this happens, RiverWare will issue a warning message but will not abort the run.

14.1.7 Previous Elevation

Three user methods are available for this method category: None, Input Elevation, and Storage Elevation Interpolation. These methods are dependent upon the selection of the Single Computed Outflow in the Solution Type category.

14.1.7.1 None

This is the default method. It performs no calculations. There are no slots specifically associated with this method.

14.1.7.2 Input Elevation

This method instantiates the Previous Groundwater Elevation slot for user input.

Slots Specific to this Method

☞ **Previous Groundwater Elevation**

Type: SeriesSlot
Units: LENGTH
Description: user input previous groundwater elevation
Information:
I/O: Required input
Links: Linked to the Previous Groundwater Elevation slot on the Agg Diversion Site or Water User object.

14.1.7.3 Storage Elevation Interpolation

This method instantiates the Previous Groundwater Elevation and Storage Elevation Table slots. User input is required in the Storage Elevation Table relating storage volume to groundwater elevation. The Previous Groundwater Elevation slot is set based on table interpolation of storage values at the previous timestep. The Previous Groundwater Elevation slot is linkable to Agg Diversion Site and Water User objects.

Slots Specific to this Method

☞ **Storage Elevation Table**

Type: TableSlot
Units: VOLUME VS. LENGTH
Description: user input relating storage volume and groundwater elevation
Information:
I/O: Required input
Links: Not linkable

☞ **PREVIOUS GROUNDWATER ELEVATION**

Type: SeriesSlot
Units: LENGTH
Description: previous groundwater elevation based upon the Storage Elevation Table
I/O: Output only
Links: Linked to the Previous Groundwater Elevation slot on Agg Diversion Site or Water User object.

14.1.8 Impulse Response Components

The Impulse Response Components category is only visible when users have selected the Single Computed Outflow in the Solution Type category and the Impulse Response method in the GW Outflow Calc category. Two user methods are available with the Impulse Response Components category: “None” and “Multiple Response Components.”

14.1.8.1 None

This is the default method. It performs no calculations. There are no slots specifically associated with this method.

14.1.8.2 Multiple Response Components

This method calculates the Response Components on the Groundwater Storage object.

Slots Specific to this Method

☞ **Response Components**

Type: NoComputeMultiSlot
Units: FLOW
Description: the slot contains the individual response components
Information: The subslots of this multislot contain the individual responses. A subslot is created when the user links this slot to a destination object.
I/O: Output only
Links: Linked to the response destination objects

☞ **Number of Response Coeffs**

Type: TableSlot
Units: NONE
Description: the integer number of response coefficients to be used for each response component
Information: A column will be added automatically for each of the responses. The name of the column is used to match it with the appropriate subslot in the Response Components slot. Each column contains a value that is the number of response coefficients to be used for that particular response.
I/O: Required Input
Links: Not linkable

☞ **Response Coeffs**

Type: TableSlot
Units: NONE
Description: the impulse response coefficients

Information: The number of coefficients must be input in each column as the value in the corresponding column in the Number of Response Coeffs slot. The columns are added and named automatically as a new subslot is added to the Response Components slot.
I/O: Required input
Links: Not linkable

As users create and remove links to the Response Components slot, corresponding subslots are added and removed. The creation of a link automatically adds a new column to the Number of Response Coeffs slot and the Response Coeffs slot to represent that particular response component. The first column on the Response Components multislot is the multislot itself and is not one of the subslots. It is set as the sum of all the subslot values. It is the responsibility of the user to make sure the data entered into the table columns correspond to the proper subslot in the Response Components slot.

$$\text{Response Components(i)}_t = C(i)_0 \text{ Impulse}_t + C(i)_1 \text{Impulse}_{t-1} + \ldots \\ + C(i)_{ncoeff-1} \text{Impulse}_{t-ncoeff-1} + C(i)_{ncoeff} \text{Impulse}_{t-ncoeff}$$

where i is an iteration variable that takes on the column value for each of the Response Components.

Each of the response components is actually a part of the total Outflow. However, because the user may not want to model all of the response components, the sum of the responses does not have to equal the Outflow.

In order to avoid mass balance issues, RiverWare does not allow both the Outflow slot and the Response Components slot to be linked. The one exception is that the Response Components slot can be linked to a Data Object if the Outflow slot is linked. This allows users to view the responses without actually routing them.

14.1.9 Negative Outflow Adjustment

The Negative Outflow Adjustment category is only visible when the user has selected the Single Computed Outflow in the Solution Type category and Impulse Response method from the Groundwater Outflow category. The category contains four methods: “None,” “Adjust Outflow,” “Adjust Components,” and “Adjust Outflow and Components.”

14.1.9.1 None

This is the default method. It performs no calculations. There are no slots specifically associated with this method.

14.1.9.2 Adjust Outflow

This method changes any negative Outflow values to zero. There are no slots specifically associated with this method.

14.1.9.3 Adjust Components

This method changes any negative Response Components values to zero. There are no slots specifically associated with this method.

14.1.9.4 Adjust Outflow and Components

This method changes any Outflow and Response Components values that are negative to zero. There are no slots specifically associated with this method.

14.1.10 Groundwater Evapotranspiration

This category contains three methods: None, Input ET Rate, and Elevation ET Table.

14.1.10.1 None

This is the default method. It performs no calculations. There are no slots specifically associated with this method. Evapotranspiration is assumed to be zero.

14.1.10.2 Input ET Rate

The Input ET Rate method allows the user to input the ET rate. This value is converted to a volume based on the Riparian Area and the timestep length. The volume of ET is used to compute storage.

☞ **ET RATE**

Type: Series Slot
Units: VELOCITY
Description: The evapotranspiration rate in units of velocity
Information: The ET Rate is used to compute the evapotranspiration volume. The ET volume is used to compute storage via the mass balance equation.
I/O: Required input
Links: Usually not linked

☞ **ET VOLUME**

Type: Series Slot
Units: VOLUME
Description: The ET Rate multiplied by the Riparian Area and the timestep length
Information: Used to compute storage
I/O: Generally output; if input, it overrides the value computed using the ET Rate
Links: Not linkable

☞ **RIPARIAN AREA**

Type: Series Slot
Units: AREA
Description: The horizontal area of the riparian zone
Information: Used to compute ET Volume
I/O: Required Input
Links: Not linkable

14.1.10.3 Input ET Volume

The Input ET Volume method allows the user to input the ET Volume. The volume of ET is used to compute storage.

☞ **ET VOLUME**

Type: Series Slot
Units: VOLUME
Description: The volume of water lost from the groundwater object through riparian evapotranspiration
Information: Used to compute storage
I/O: Input only
Links: Not linkable

14.1.10.4 Elevation ET Table

The Elevation ET Table method computes evapotranspiration given the relationship between aquifer elevation and ET rate. The ET rate found in the Elevation ET Table is multiplied by the Riparian Area to get the ET Volume.

☞ **ELEVATION ET TABLE**

Type: Table Slot
Units: LENGTH VS VELOCITY
Description: A table relating water table elevation to ET rate
Information: The ET rate is determined based on the beginning of timestep (previous) elevation value. The ET rate is then multiplied by the Riparian Area and timestep length to compute the ET Volume.
I/O: Required Input
Links: Not linkable

☞ **ET VOLUME**

Type: Series Slot
Units: VOLUME
Description: The ET Rate multiplied by the Riparian Area and the timestep length
Information: Used to compute storage
I/O: Output; if input, it overrides the value computed from the Elevation ET Table
Links: Not linkable

☞ **RIPARIAN AREA**

Type: Series Slot
Units: AREA
Description: The horizontal area of the riparian zone
Information: Used to compute ET Volume
I/O: Required Input
Links: Not linkable

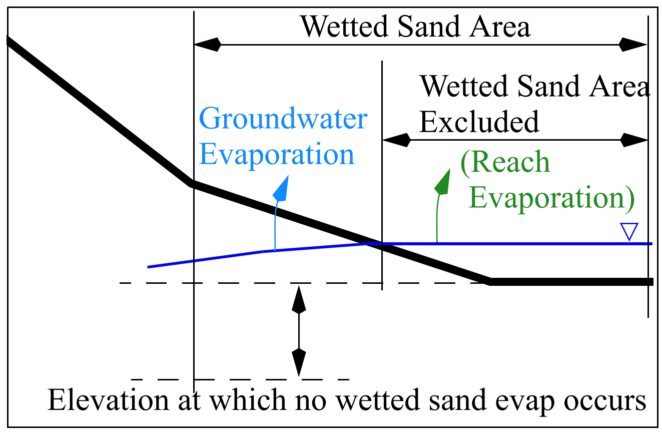
14.1.11 Groundwater Evaporation

This category contains two methods: None and Wetted Sand Evaporation. These methods are only available when the **Connected Groundwater Object** method is selected in the **Solution Type** category.

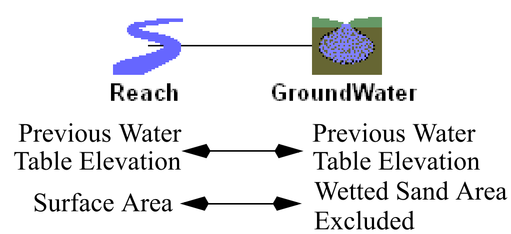
14.1.11.1 None

This is the default method. It performs no calculations. There are no slots specifically associated with this method. Evaporation is assumed to be zero.

14.1.11.2 Wetted Sand Evaporation



This method computes the evaporation that occurs from the wetted sand of the shallow groundwater. As shown in the half cross section figure to the right, the groundwater evaporation comes from the wetted sand in the shallow groundwater that is adjacent to the water surface, but optionally, does not include the water surface. The area that contributes to the Evaporation is the "Wetted Sand Area" minus the "Wetted Sand Area Excluded". The final evaporation is then computed using this reduced area, a coefficient, and the evaporation rate. The evaporation rate decreases from the pan evaporation rate at the surface to zero at some depth. This decrease in the rate of evaporation is simulated by a table that relates the elevation to the percent of the Pan Evaporation Rate at that elevation. Also, if there is no flow in the reach and the elevation is below a certain threshold level, the evaporation is limited to a reduced periodic rate.



This method was designed to work with a connected Reach object where the reach computes its Surface Area (**HERE (Section 22.1.19.3)**) and then propagates the value to the "Wetted Sand Area Excluded". In this linking approach, evaporation from the water surface comes from the Reach while evaporation from the wetted sand comes out of the groundwater.

Slots Specific to this Method

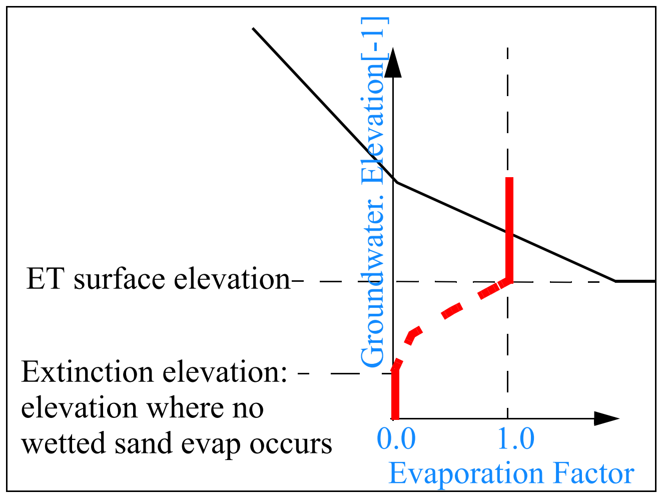
☞ Evaporation

Type: Series Slot

Units: VOLUME
Description: This slot is the output to the method. It holds the losses as calculated by the equation.
Information:
I/O: Output only
Links: Not linkable

☞ ELEVATION EVAPORATION FACTOR TABLE

Type: Table
Units: LENGTH VS DECIMAL
Description: This table relates the groundwater elevation with a factor that is used to compute the fraction of pan evaporation that occurs at that elevation.
Information: The values should be between 0.0 and 1.0 (0% to 100%), inclusive. The factor should be zero at (and below) the elevation at which no evaporation will occur. It should be 1.0 at (and above) the ET surface elevation (typically the streambed elevation of the reach above). See the diagram and the table for a view of what this table should represent. Note, the Elevations must be in increasing order.



I/O: Required Input
Links: NA

	Data to enter in the slot	
	Elevation	Pan Evaporation Factor
	5460	0.0
Elevation at extinction depth	5490	0.0
	5491	0.2
	5492	0.5
	5493	0.8
	5494	0.9
ET surface elevation	5495	1.0
Full Channel Elevation	5500	1.0

Pan Coefficient

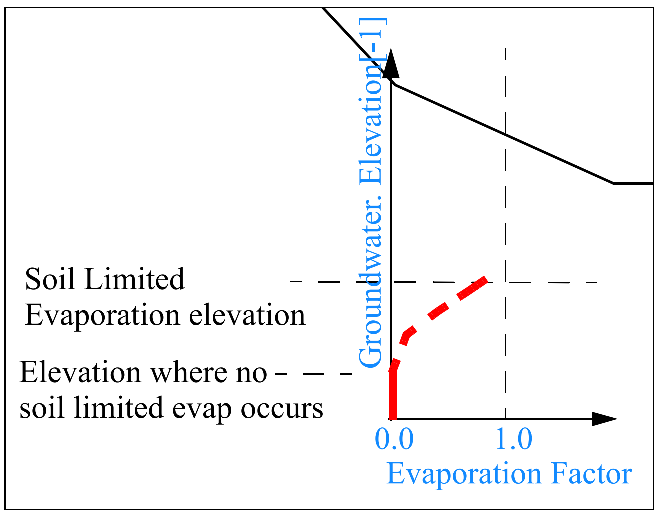
Type: Periodic
Units: No Units
Description: This slot holds a coefficient that is applied to the computation.
Information: The coefficient should be between 0 and 1.0.
I/O: Required Input
Links: NA

Pan Evaporation

Type: Series
Units: Velocity (Length/Time)
Description: This slot contains the pan evaporation rate.
Information: This slot must have a valid input or an error will occur.
I/O: Input only
Links: Can be linked

Soil Limited Evaporation Factor Table

Type: Table
Units: Length vs Decimal
Description: This table relates the groundwater elevation with a factor that is used to compute the fraction of soil limited evaporation that occurs at that elevation.
Information: The values should be between 0.0 and 1.0 (0% to 100%), inclusive. The factor should be zero at (and below) the elevation at which no evaporation will occur. The largest factor should be at the Soil Limited Evaporation Elevation. See the diagram and the table for a view of what this table should represent. Note, the Elevations must be in increasing order.



I/O: Required Input if Soil Limited Evaporation Elevation is specified
Links: NA

	Data to enter in the slot	
	Elevation	**Soil Limited Evaporation Factor**
Elevation where no soil limited evap occurs	5490	0.0

	Data to enter in the slot	
	Elevation	Soil Limited Evaporation Factor
	5495	0.5
Soil Limited Evaporation Elevation	5495.5	0.9

Soil Limited Evaporation Rate

Type: Periodic
Units: Velocity (Length/Time)
Description: The evaporation rate when there is no area to be excluded and the water table elevation is below the Soil Limited Evaporation Elevation
Information:
I/O: Input required when Soil Limited Evaporation Elevation is specified
Links: NA

Soil Limited Evaporation Elevation

Type: Scalar
Units: Length (elevation)
Description: The elevation at which evaporation is only limited by the soil above, not the pan evaporation rates.
Information: If this slot is not specified, then the method will always use the second equation below; Soil Limiting Evaporation will not be modeled.
I/O: Optional Input
Links: No

Wetted Sand Area

Type: Scalar
Units: Area
Description: The horizontal area of the aquifer that is considered wetted and could cause evaporation.
Information: This area includes any surface area of connected reaches
I/O: Input Only
Links: NA

Wetted Sand Area Excluded

Type: Series Slot
Units: Area
Description: The wetted sand area that should not be included in the computation
Information: This provides a slot to adjust the wetted surface area based on current conditions.
I/O: If not input, set by a rule, or linked, it is assumed to be zero.
Links: Linkable. Typically, this slot is linked to the Surface Area on the Reach directly above this groundwater object

This method will be executed from the **solveGWMB_givenPreviousElevations** dispatch method before mass balance takes place. First, if the **Wetted Sand Area Excluded** is linked but not valid, the Groundwater object will exit the dispatch method and wait for the connected object to compute a value. Otherwise, if it is not valid, the **Wetted Sand Area Excluded** will default to 0 (but will not be set).

If the **Wetted Sand Area Excluded** is greater than or equal to the **Wetted Sand Area**, the **Evaporation** is set to zero and the method stops executing. This meets the need that wetted sand evaporation does not occur if the channel is full.

If the **Soil Limited Evaporation Elevation** is valid, the Elevation Previous is less than or equal to **Soil Limited Evaporation Elevation,** and the **Wetted Sand Area Excluded** is equal to zero. Then the previous **Elevation** is looked up on the **Soil Limited Evaporation Rate** periodic slot to get the soil limited evaporation rate that would occur on that date. This is stored as a temporary variable tempSoilLimitedEvaporationRate. Also, the previous **Elevation** is looked up on the **Elevation Evaporation Factor Table** to get the fraction of **Soil Limited Evaporation Rate** that occurs at that elevation. This is stored as a temporary variable tempSoilLimitedEvapFactor. The evaporation (flow) is computed as:

$$EvaporationFlow[t] = tempSoilLimitedEvaporationRate \times tempSoilLimitedEvapFactor \times WettedSandArea$$

When the above conditions are not true, the previous **Elevation** is looked up on the **Elevation Evaporation Factor Table** to get the fraction of pan evaporation that occurs at that elevation. This is stored as a temporary variable tempElevEvapFactor. Then, the evaporation (flow) is computed as:

$$EvaporationFlow[t] = PanEvaporation[t] \times tempElevEvapFactor \times PanCoefficient[t] \times (WettedSandArea - WettedSandAreaNotIncluded[t])$$

Finally, the Evaporation Flow is converted to a volume and saved as a local Evaporation variable.

The method finishes and exits. The dispatch method sets the **Evaporation** slot and subtracts it as a loss from the groundwater mass balance computation.

14.1.12 Lateral Link Direction

The “Lateral Link Direction” category is dependent upon selection of the “Head Based Groundwater Grid” or the “Link to MODFLOW GW” method. When the “Head Based Groundwater Grid” method is selected, the user needs to specify the number and location of the connected objects with respect to the current object. This category contains 15 user methods which are all the combinations of Upstream, Downstream, Left and Right. For example a groundwater object may be connected to only one other groundwater object which is upstream. Or it may be connected to four other groundwater objects which are upstream, downstream, left, and right. These methods are used to control the slots added and the dispatch conditions used (the dispatch conditions will depend on the Elevation Upstream/Downstream/Left/Right Previous slots).

When the “Link to MODFLOW GW” method is selected, the user needs to specify the number and location of the connected objects with respect to the current object. This category contains 3 user methods Upstream, Downstream, Upstream and Downstream. For example, a groundwater object may be connected to only one other groundwater object which is upstream. Or it may be connected to both an upstream and downstream object. These methods are used to control the slots added and the dispatch conditions used (the dispatch conditions will depend on the Elevation Upstream and Downstream Previous slots).

14.1.12.1 No Linked Objects

A **No Linked Objects** method is also available. This method can be used without specifying connections to other groundwater objects.

14.1.12.2 Upstream, Downstream, Left, and/or Right

NOTE: For the remainder of this section, the term Elevation U/D/L/R Previous slots will be used to represent any or all of the combinations of the four slots: Elevation Upstream Previous, Elevation Downstream Previous, Elevation Left Previous, and Elevation Right Previous.

☞ **ELEVATION UPSTREAM, DOWNSTREAM, LEFT, AND RIGHT PREVIOUS**

Type: Series Slot
Units: LENGTH
Description: The previous elevation of a connected groundwater object
Information: These are actually 4 separate slots. The user is allowed to choose the number and the location of the connected groundwater objects (with respect to the current object). There is a slot for each connected object. These slots were previously called “Previous Adjacent Elevation Left/Right/Upstream/Downstream”
I/O: Output only
Links: These slots are linked to the Elevation Previous slots on each of the connected groundwater objects

☞ Conductance Upstream, Downstream, Left, and Right

Type: Scalar Slot
Units: AREA PER TIME
Description: The conductance values that correspond to the area between this groundwater object and each of the connected groundwater objects
Information: These are 4 separate slots. The user is allowed to choose the number and location of the connected groundwater objects. There is a slot for each connected object.
I/O: Input or computed as specified HERE (Section 14.1.5).
Links: Not linkable

☞ Flow Upstream, Downstream, Left, Right

Type: Series Slot
Units: FLOW
Description: The flow between the current object and each of the connected groundwater objects
Information: These are 4 separate slots. The user is allowed to choose the number and location of the connected groundwater objects. There is a slot for each connected object. A positive value represents an inflow to the groundwater object.
I/O: Output only
Links: Not linkable

☞ Flow Factor Upstream, Downstream, Left Right

Type: Series Slot
Units: FRACTION
Description: Proportion of the flow initially calculated from the previous elevations that will actually occur
Information: This value is between 0 and 1. It can remain NaN, and the value is assumed to be 1.0. If the slot is not linked, it is unused (assumed to be 1). If the slot is linked, and the calculated lateral flow would result in a negative storage, the dispatch method sets this slot to a value less than 1 in order to reduce the lateral flow and prevent negative storage.
I/O: Output only
Links: Yes, linkable to other Flow Factor Left/Right/Upstream/Downstream slots. For example, the Flow Factor Right will be linked to the Flow Factor Left on an adjacent object. RiverWare requires that either all of the Flow Factor slots that are in use in a groundwater network are linked or that none are linked. Otherwise a gain or loss of mass could occur. Linking some Flow Factor slots and not others within a groundwater network will cause the run to abort with an error message.

In the code, one method deals with all 15 user methods in the Lateral Link Direction category. This method is called by the dispatch method and computes temporary values for the Flow slots. These temporary values may be adjusted in the dispatch method (HERE (Section 14.2.2)) by applying the corresponding Flow Factor. Depending on the number of connected objects, the method performs one or all of the following calculations:

$$\text{temp Flow Upstream} = \text{Conductance Upstream} \times (\text{Elevation Upstream Previous} - \text{Elevation Previous})$$

$$\text{temp Flow Downstream} = \text{Conductance Downstream} \times (\text{Elevation Downstream Previous} - \text{Elevation Previous})$$

$$\text{temp Flow Left} = \text{Conductance Left} \times (\text{Elevation Left Previous} - \text{Elevation Previous})$$

$$\text{temp Flow Right} = \text{Conductance Right} \times (\text{Elevation Right Previous} - \text{Elevation Previous})$$

The results are passed back to the dispatch method and used in the mass balance equation to solve for storage. The actual slot values are set from the dispatch method.

Note: The Flow terms are computed based on the previous timestep elevation differences. This is done to avoid iteration problems that would result if the groundwater flows were computed based on current timestep elevations.

Note: Mass may be lost from the system if the conductance values are not identical on either side of a given link. For example in a two aquifer system where an upstream GW Object 1 is linked to a downstream GW Object 2, the upstream conductance on GW Object 2 must be identical to the downstream conductance on GW Object 1 to conserve mass. If you use the Compute Conductance method, HERE (Section 14.1.5.2), on each groundwater object, then adjacent conductances are guaranteed to be identical.

14.2 Dispatch Methods

14.2.1 solveGWMB_givenInflow

This dispatch method is used for Single Computed Outflow calculations only. With a given Inflow, the dispatch method can execute to calculate Outflow and Storage. The required knowns and unknowns (at the current timestep) are listed below.

Required Knowns

- Inflow

Required Unknowns

- Outflow
- Storage

The solveGWMB_givenInflow dispatch method is very simple. First the previous timestep's Storage is checked. If it is not known, the method is exited because a previous Storage value is required for any of the user methods in the **Groundwater Outflow** category to execute successfully. If there is a previous Storage value, the dispatch method executes the selected GW Deep Percolation method and the selected **Groundwater Outflow** method. If the selected **Groundwater Outflow** method is ImpulseResponseOutflow, the dispatch method next executes the selected Impulse Response Components method and Negative Outflow Adjustment method.

A local variable, storage flow, is calculated as the previous timestep's Storage value divided by the length of the current timestep. This represents the flow rate required to drain the aquifer of all its stored water in the current timestep. If the calculated Percolation plus Outflow is greater than storage flow, a warning is posted and the following calculations take place.

Outflow is recalculated by the following equation:

$$\text{Outflow} = \frac{\text{Outflow}}{(\text{Percolation} + \text{Outflow}) / (\text{storage flow } (-1))}$$

However, if Input Percolation is selected in the GW Deep Percolation category, Outflow is recalculated as:

$$\text{Outflow} = \text{storage flow} - \text{Percolation}$$

Percolation, if not input, is recalculated by the following equation. A warning is posted to notify the user of this change.

$$\text{Percolation} = \frac{\text{Percolation}}{(\text{Percolation} + \text{Outflow}) / (\text{storage flow (-1)})}$$

This prevents the storage from going negative. If there is no method selected to calculate Percolation, it is assumed to be zero.

The Groundwater Pumping method is then executed to compute Available For Pumping at the current timestep based on the previous Storage and the current Outflow and Percolation.

Percolation (if applicable), Outflow, and Pumped Flow (if applicable) are then converted to volumetric values by multiplying them by the length of the timestep. The following mass balance is then performed to calculate the Storage:

$$\text{Storage(t)} = \text{Storage(t-1)} + \text{volume of Inflow} - \text{volume of Outflow} \\ - \text{volume of Percolation} - \text{volume of Pumped Flow}$$

14.2.2 solveGWMB_givenPreviousElevations

This dispatch method is used for Head Based Groundwater Grid calculations only. The required knowns and unknowns (at the current timestep) are listed below.

Required Knowns

☞ **Inflow From Surface Water**

☞ **Elevations Left/Right/Upstream/Downstream Previous**

There is a different set of dispatch conditions for each of the methods in the Lateral Link Direction category. If the **No Linked Objects** method is selected, then the elevations slots are not applicable.

Required Unknowns

☞ **Elevation**

☞ **Storage**

The solveGWMB_givenPreviousElevations dispatch method will compute the current Storage and Elevation values based on the mass balance equation described below. Following is a list of the steps performed in the dispatch method.

Conditions for dispatching:

- The dispatch method will execute when the Inflow From Surface Water and the Elevation Upstream/Downstream/Right/Left Previous slots are known. Depending on the number of connected groundwater objects, one or all of the Elevation Upstream/ Downstream/Right/Left Previous slots may be active.

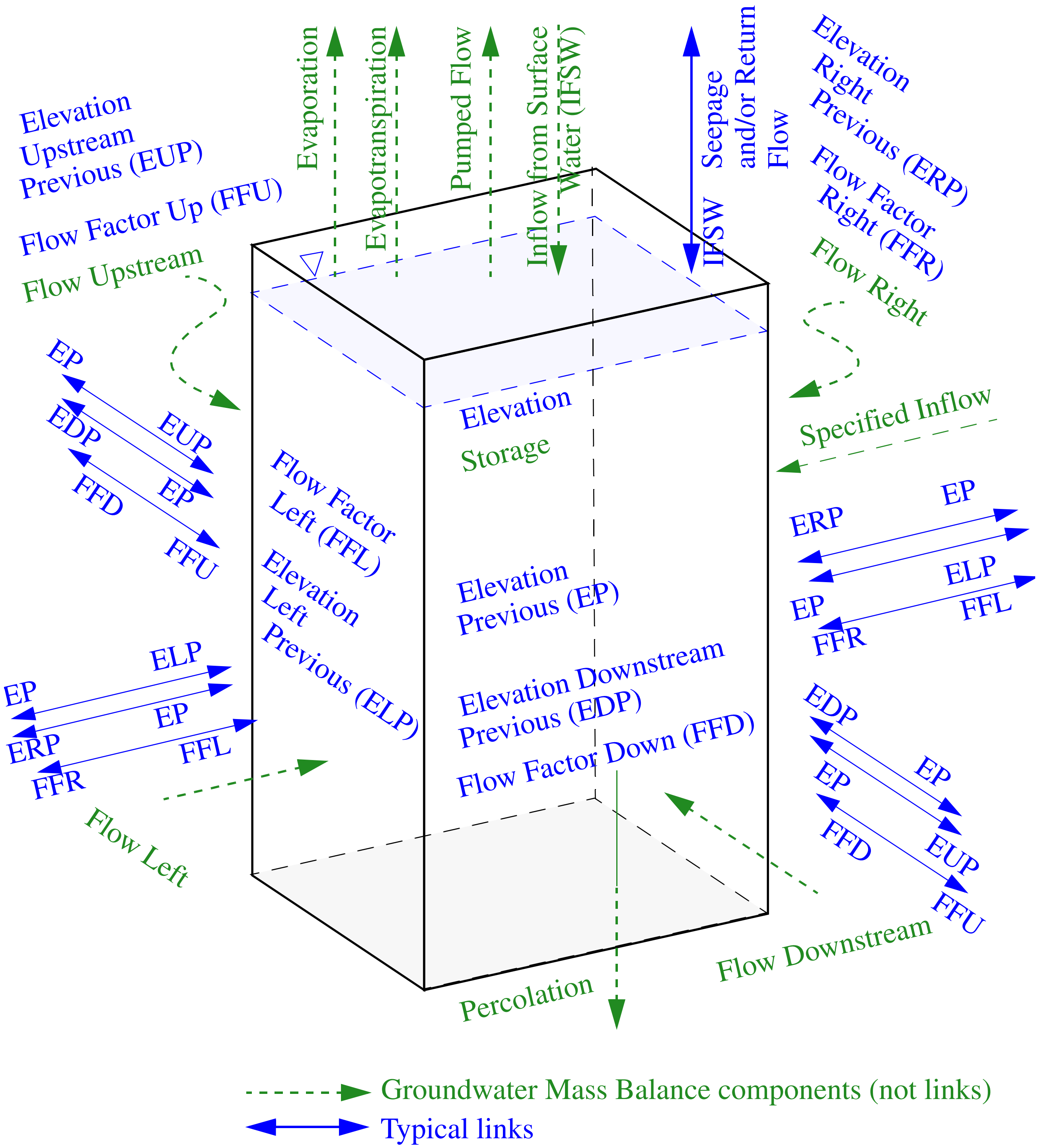


Diagram of the Groundwater Storage object including the mass balance components and typical links

- If the Inflow From Surface Water slot is not being used (is not linked) it is set to zero for all timesteps at the beginning of the run.

- Whenever a new value is received for Inflow From Surface Water or any of the Elevation Previous slots, the object will re-dispatch.

Steps in the dispatch method:

NOTE: For the remainder of this section, the term U/D//R/L will be used to represent a slot or variable for any or all of the combinations of the four directions, for example Flow U/D/R/L to represent Flow Upstream, Flow Downstream, Flow Right and Flow Left. The set of directions that are actually used depends on the method selected in the Lateral Link Direction category.

- If the previous Storage is not known, the method exits and waits. In the beginning of run, there is a check to ensure that the user inputs an initial storage value and initial Elevation value.
- A *tempStorage* local variable is set equal to the previous timestep Storage. This is updated at each step by the inflows and outflows to track the running mass balance.

$$tempStorage = \text{Storage}[t-1]$$

- **Get Preliminary Lateral Flows:** A method is called to compute the preliminary, unadjusted lateral flows, temp Flow Upstream/Downstream/Right/Left. One or all of these slots may be computed based on the number of connected groundwater objects. For any Flow slot that is not in use, the corresponding temp Flow value gets set to zero.

$$\text{temp Flow U/D/R/L} = \text{Conductance U/D/L/R} \times (\text{Elevation U/D/L/R Previous} - \text{Elevation Previous})$$

- **Add Lateral Inflows:** All of the positive lateral flows (lateral inflows) are added to the running mass balance. The corresponding Flow Factor is applied. If the Flow Factor is NaN, it is assumed to be 1.0, but it is not set. If the Flow Factor is not linked, it is unused and is assumed to be 1.0.

  FOR (U/D/R/L)

  IF (temp Flow U/D/R/L > 0)

$$\text{temp Flow U/D/R/L} = \text{temp Flow U/D/R/L} \times \text{Flow Factor U/D/R/L}$$

$$tempStorage = tempStorage + \text{temp Flow U/D/R/L} \times timestepLength$$

  END IF

- **Add Inflow from Surface Water:** Inflow from Surface Water is always used as is. If the value is negative and causes the *tempStorage* to become negative, a warning will be issued.

$$tempStorage = tempStorage + \text{Inflow from Surface Water} \times timestepLength$$

- **Add Specified Inflow:** Specified Inflow is always used as is. If the value is negative and causes the *tempStorage* to become negative, a warning will be issued.

$$tempStorage = tempStorage + \text{Specified Inflow} \times timestepLength$$

- **Subtract Pumped Flow:** If the Input Pumped Flow method is selected, and the Pumped Flow slot is linked to another object(s) but does not contain a value, the method exits and waits for the linked water user object(s) to set Supplemental Diversion. These values are then passed back to the groundwater object through the Pumped Flow slot. If the Pumped Flow value is already present, then it is subtracted from the running mass balance. If no Groundwater Pumping method is selected, then Pumped Flow is not included in the mass balance.

$$tempStorage = tempStorage - \text{Pumped Flow} \times timestepLength$$

- **Subtract Percolation:** If the GW Deep Percolation method is selected, then it is called. Percolation is initially computed as discussed in the Head Based Percolation user method section **HERE (Section 14.1.4.5)**. If Percolation is specified (input or set by rules), then it is always used as is. If the value is positive and causes the *tempStorage* to become negative, a warning will be issued. Otherwise if Percolation is not specified, it is checked that it will not cause the *tempStorage* to become negative. Positive Percolation is flow out of the groundwater object.

  IF Percolation is input or set by rules

  Use Percolation as is

  ELSE

  Set the Percolation slot, but limit it to the amount of water remaining in *tempStorage*. Do not allow storage to become (more) negative.

  IF (*tempStorage* <= 0)

$$\text{Percolation} = Min(\text{Percolation}, 0)$$

  ELSE

$$\text{Percolation} = Min\left(\text{Percolation}, \frac{tempStorage}{timestepLength}\right)$$

  END IF

  END IF.

$$tempStorage = tempStorage - \text{Percolation} \times timestepLength$$

- **Subtract Evaporation:** The Groundwater Evaporation method is executed. If the **Wetted Sand Area Excluded** is visible and linked but not valid, the Groundwater object exits the dispatch method and waits for the connected object to compute a value. Otherwise, evaporation is incorporated into the running mass balance with a check that it will not cause *tempStorage* to become negative.

  IF (*tempStorage* <= 0)

$$\text{Evaporation} = 0$$

ELSE

$$\text{Evaporation} = Min(\text{Evaporation}, tempStorage)$$

END IF

$$tempStorage = tempStorage - \text{Evaporation}$$

- **Subtract Evapotranspiration:** The Groundwater Evapotranspiration method is executed and calculates a preliminary ET Volume. If ET Volume is specified (input or set by rules), then it is always used as is. If the value causes the *tempStorage* to become negative, a warning will be issued. Otherwise if ET Volume is not specified, it is checked that it will not cause the *tempStorage* to become negative.

IF ET Volume is input or set by rules

Use ET Volume as is

ELSE

Set the ET Volume slot, but limit it to the amount of water remaining in *tempStorage*. Do not allow storage to become (more) negative.

IF (*tempStorage* <= 0)

$$\text{ET Volume} = Min(\text{ET Volume}, 0)$$

ELSE

$$\text{ET Volume} = Min\left(\text{ET Volume}, \frac{tempStorage}{timestepLength}\right)$$

END IF

END IF.

$$tempStorage = tempStorage - \text{ET Volume}$$

- **Incorporate Lateral Outflows:** A preliminary storage value is calculated after incorporating all negative lateral flows (lateral outflows). If that value is negative and the Flow Factor U/D/R/L slots are linked, a Flow Factor is calculated that will result in a storage of 0. If the *tempStorage* before applying the lateral outflows was already negative, then the Flow Factor will be zero. The Flow Factor will propagate across the link to be applied to the corresponding lateral inflow on the linked groundwater object. If the resulting preliminary storage is non-negative, then the Flow Factor is assumed to be 1.0, but the slot value is not set. If the Flow Factor U/D/R/L slots are not linked, they will not get used, and thus any negative preliminary storage will remain as the final storage.

IF (*tempStorage* < 0)

IF (temp Flow U/D/R/L < 0)

$$\text{Flow Factor U/D/R/L} = 0$$

$$\text{temp Flow U/D/R/L} = 0$$

END IF

ELSE

FOR (U/D/R/L)

IF (temp Flow U/D/R/L < 0)

$$lateralFlowOut = lateralFlowOut + \text{temp Flow U/D/R/L}$$

END IF

END FOR

$$tempStorageWithLateralFlow = tempStorage + lateralFlowOut \times timestepLength$$

IF (*tempStorageWithLateralFlow* < 0)

$$flowFactor = 1 - \frac{tempStorageWithLateralFlow}{lateralFlowOut \times timestepLength}$$

IF (temp Flow U/D/R/L < 0 AND Flow Factor U/D/R/L is linked)

$$\text{temp Flow U/D/R/L} = \text{temp Flow U/D/R/L} \times flowFactor$$

$$\text{Flow Factor U/D/R/L} = flowFactor$$

END IF

END IF

END IF

Note: This logic assumes that either none of the Flow Factor slots are linked, and thus the Flow Factor slots are not used, or all of the Flow Factor slots are linked appropriately to Flow Factor slots on adjacent Groundwater objects. If the linking of the Flow Factor slots is inconsistent, it could result in a gain or loss of mass.

Incorporate the final negative lateral flows into the running mass balance.

FOR (U/D/R/L)

IF (temp Flow U/D/R/L < 0)

$$tempStorage = tempStorage + \text{temp Flow U/D/R/L} \times timestepLength$$

END IF

END FOR.

- The final flow values are then set on the Flow U/D/R/L slots.

$$\text{Flow U/D/R/L} = \text{temp Flow U/D/R/L}$$

- The final Storage slot value is computed by the following mass balance equation:

$$\begin{aligned}\text{Storage} = {} & \text{Storage(t-1)} + \text{Flow Upstream} + \text{Flow Downstream} \\ & + \text{Flow Left} + \text{Flow Right} + \text{Inflow From Surface Water} \\ & - \text{ET Volume} - \text{Percolation} - \text{Pumped Flow} - \text{Evaporation} + \text{Specified Inflow}\end{aligned}$$

The groundwater flow, percolation, Inflow from Surface Water, Specified Inflow and pumped flow terms in the above equation are actually the volumes associated with those flow rates (flow rate multiplied by timestep length). Some of the terms may not be active depending on method selection (i.e. ET Vol, Percolation, Pumped Flow).

If Storage is negative, a warning will be issued but the run will not abort.

- The Elevation slot is computed as follows:

$$\text{Elevation (t)} = \text{Elevation (t-1)} + \frac{\text{Storage (t)} - \text{Storage (t-1)}}{\text{Specific Yield} \times \text{Aquifer Area}}$$

- The Elevation Previous slot is set at the next timestep given the current Elevation value. This will propagate to the Elevation Left/Right/Upstream/Downstream Previous slots on the connected groundwater objects. This will trigger the connected groundwater objects to dispatch at the next timestep.
- The Previous Storage slot is set at the next timestep given the current Storage value. This is an invisible slot. It will trigger the current groundwater object to re-dispatch at the next timestep if necessary.
- If the Input Pumped Flow method is selected, and the Available For Pumping slot is linked to another object(s), Available For Pumping for the next timestep is computed from the current Storage with ET Volume and Percolation removed if they are already known. This value will then propagate to the linked water user objects which can then compute Supplemental Diversion on the next timestep. These values are then passed back to the groundwater object through the Pumped Flow slot.

$$\text{Available for Pumping}[t+1] = Max(\text{Storage as flow}[t] - \text{Percolation}[t+1] - \text{ET Vol as flow}[t+1], 0)$$

Note: Mass Balance - The groundwater flows between the objects are only used to compute storage. So each individual object will mass balance, but the flow values are not transferred between objects. This is because the only linked slots are the Elevation Previous slots and Flow Factor slots. For these objects to mass balance, and for the groundwater system to mass balance, the conductance values must be identical for connected groundwater objects on either side of a given link. If you use the Compute

Conductance method, HERE (Section 14.1.5.2), on each groundwater object, then adjacent conductances are guaranteed to be identical.

14.2.3 solveGWInflows_givenElevations

This dispatch method is used for Head Based Boundary Condition solution type only. The purpose of this solution type is allow for linking of a groundwater network to a Reservoir. The required knowns and unknowns (at the current timestep) are listed below.

REQUIRED KNOWNS

☞ **ELEVATION PREVIOUS**

☞ **ELEVATION PREVIOUS LEFT/RIGHT/UPSTREAM/DOWNSTREAM**

There is a different set of dispatch conditions for each of the methods in the Lateral Link Direction category. If the **No Linked Objects** method is selected, then the **Elevation Left/Right/Upstream/Downstream Previous** slots are not applicable.

REQUIRED UNKNOWNS

☞ **INFLOW FROM SURFACE WATER**

Steps in the dispatch method:

- The **Elevation** slot is set for the previous timestep:

$$\text{Elevation}[t-1] = \text{Elevation Previous}[t]$$

The **Elevation Previous** value will typically be propagated across a link from the Reservoir **Previous Elevation** slot.

- A method is called to compute **Flow Upstream**, **Downstream**, **Left**, and **Right**. One or all of these slots may be computed based on the number of connected Groundwater objects. These are computed based on the **Elevation Previous** and **Elevation Left/Right/Upstream/Downstream Previous** and **Conductance Left/Right/Upstream/Downstream**. The **Flow Factor Left/Right/Upstream/Downstream** is applied if it is not NaN. (The **Flow Factor** will be calculated by the linked Groundwater object using the **Head Based Groundwater Grid** solution type. If it is NaN, it is assumed to be 1.)

$$\text{Flow U/D/R/L} = \text{Conductance U/D/L/R} \times (\text{Elevation U/D/L/R Previous} - \text{Elevation Previous}) \times \text{Flow Factor U/D/R/L}$$

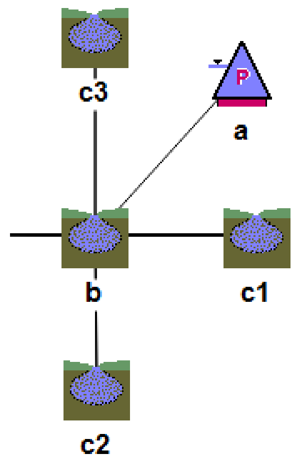
- If a **Deep Percolation** method is selected, then it is called. **Percolation** is computed as discussed in the user method sections for the **Deep Percolation** category.

- Finally the mass balance is performed to compute the **Inflow from Surface Water**:

$$\text{Inflow From Surface Water} = -(\text{Flow Upstream} + \text{Flow Downstream} + \text{Flow Right} + \text{Flow Left}) + \text{Percolation}$$

Reservoir/Groundwater Object Interaction

Following is a description of the typical Groundwater object and Reservoir interaction. It consists of three sets of objects



- Reservoir with **Linked Seepage** method (**a**)
- One Groundwater object with the **Head Based Boundary Condition** method (**b**)
- One to four laterally linked Groundwater objects solving using **Head Based Groundwater Grid** (**c1-c4**)

On the first timestep, the **c1-c4** get an **Inflow from Surface Water** and solve for **Storage** and **Elevation** given **Elevation Previous** values. Also, **b** solves given **Elevation Previous** which came across the link from the Reservoir as set on the Reservoir during initialization. A new **Inflow from Surface Water** is computed for **b** and propagates to the **Reservoir.Seepage**. The Reservoir dispatches once it has enough information and solves for **Storage** and **Pool Elevation** at *t* and sets **Previous Pool Elevation** at *t+1*. This propagates to **Elevation Previous** on **b** at *t+1*, which in turn propagates to **Elevation Upstream/Downstream/Right/Left Previous** on **c1-c4** at *t+1*. This allows **c1-c4** to solve at *t+1,* and **b** also solves at *t+1* for **Inflow from Surface Water**. This linked value provides **Reservoir.Seepage** for the next timestep. The process continues through the run where the **c1-c4** and **b** solve one timestep before the Reservoir is able to solve.

14.2.4 SolveGWMB_MODFLOW_Link

This dispatch method is available only when the "Link to MODFLOW GW" method is selected. The required knowns for this dispatch method depend on the specified connected groundwater objects. There are a different set of dispatch conditions for each of the methods in the "Lateral Link Direction" category.

Click HERE (Section 7.2.1) to view the RiverWare - MODFLOW Connection Functionality Guide. A description of the Groundwater object's specific data configuration is presented in that guide HERE (Section 7.2.1.3.3).

Required Knowns

☞ **Inflow From Surface Water**

☞ **Elevation Upstream/Downstream Previous**

Required Unknowns

☞ Elevation

☞ Storage

The dispatch method computes the current Storage and Elevation values based on the mass balance equation described below. Following is a list of the steps performed in the dispatch method.

Steps in the dispatch method:

- If the previous Storage is not known, the method exits and waits. In the beginning of run, there is a check to ensure that the user inputs an initial storage value and initial Elevation value.
- A method is called to compute Flow Upstream and Downstream. One or both of these slots may be computed depending on the connection of the objects.
- The GW Deep Percolation method is called. Percolation is computed.
- Storage is computed by the following mass balance equation:

$$\text{Storage} = \text{Storage(t-1)} + \text{Groundwater Flow Upstream} + \text{Groundwater Flow Downstream} \\ - \text{Lateral Flux} - \text{Percolation} + \text{Inflow From Surface Water}$$

- The groundwater flow, flux, percolation, and surface water inflow terms in the above equation are actually the volumes associated with those flow rates (flow rate multiplied by timestep length). If Storage is negative, a warning will be issued but the run will not abort.
- The Elevation slot is computed as follows:

$$\text{Elevation (t)} = \text{Elevation (t-1)} + \frac{\text{Storage (t)} - \text{Storage (t-1)}}{\text{Specific Yield} \times \text{Aquifer Area}}$$

- The Elevation Previous slot is set at the next timestep given the current Elevation value. This will propagate to the Elevation Previous slots on a the connected groundwater objects. This will trigger the connected groundwater objects to dispatch at the next timestep.
- The Previous Storage slot is set at the next timestep given the current Storage value. This is an invisible slot. It will trigger the current groundwater object to re-dispatch at the next timestep if necessary.

15. Inline Power Plant

This object models an inline power plant, otherwise known as a run-of-the-river power plant. These power facilities rely on the natural slope of the river channel to provide necessary head and assumes there is very little or no storage.

General Slots

(slots which always appear for this object)

☞ Bypass

Type: Series Slot
Units: FLOW
Description: Flow diverted around the power plant
Information: Set to the sum of Min Bypass and any excess water that the turbines cannot pass. If Turbine Release is known, will be set to (or Outflow) minus Turbine Release.
I/O: Output
Links: Not Linkable

☞ Inflow

Type: Series Slot
Units: FLOW
Description: Flow into the power plant
Information:
I/O: Input or Output
Links: May be linked to the Outflow of another object.

☞ Max Turbine Release

Type: Table Slot
Units: FLOW
Description: Maximum flow which can pass through the turbines.
Information: A single value must be input.
I/O: Required Input
Links: Not Linkable

☞ **Min Bypass**

Type: Series Slot
Units: FLOW
Description: Flow intentionally diverted around the power plant.
Information: Assumed to be zero in Bypass calculation if not user input.
I/O: Input
Links: Not Linkable

☞ **Outflow**

Type: Series Slot
Units: FLOW
Description: Flow rate below the power plant
Information: Set to the sum of Turbine Release and Bypass.
I/O: Input or Output
Links: May be linked to the Inflow of another object.

☞ **Power Plant Cap Fraction**

Type: Series Slot
Units: NONE
Description: Fraction of the capacity at which the plant is operating.
Information: Used when the plant is not operating at full capacity (e.g. a turbine in not functional)
I/O: Optional, the slot defaults to 1 if not input by the user.
Links: Not Linkable

☞ **Turbine Release**

Type: Series Slot
Units: FLOW
Description: Flow passing through the turbines.
Information: This value is used in power calculations.
I/O: Input or Output
Links: Not Linkable

15.1 User Methods

15.1.1 Inline Turbine Release and Bypass

These methods determine how Turbine Release and Bypass are specified or calculated

15.1.1.1 Specify Flows

In this default method, you can specify flows that will be used in the Turbine Release and/or Bypass computation. You must specify a Max Turbine Release value. You can then:

- Optionally, input or set Turbine Release with a rule
- Optionally, input or set Min Bypass with a rule

SLOTS SPECIFIC TO THIS METHOD: NONE

METHOD DETAILS:

This method calculates Bypass and/or Turbine Release using the following algorithm:

If the Min Bypass slot is not user input, it is assumed to be zero.

The temporary *Flow* variable is set equal to the Inflow or Outflow, depending on the dispatch method that calls this method.

Next, the method checks to see if the Turbine Release value has been input or set by rules.

If it has, Bypass is calculated as *Flow* minus Turbine Release.

$$Bypass = Flow - TurbineRelease$$

Several checks are performed to make sure that the specified Turbine Release is not greater than the Max Turbine Release and that Turbine Release plus Min Bypass is not greater than the *Flow*.

If Turbine Release is not input or set by rules, the Bypass is computed as:

$$Bypass = Max(0, \text{Min Bypass}, Flow - MaxTurbineRelease \times PowerPlantCapFraction)$$

Once the Bypass value is known, Turbine Release can be set to the Flow minus the Bypass:

$$TurbineRelease = Flow - Bypass$$

The flow used in the power method, *PowerFlow* is set equal to Turbine Release:

$$PowerFlow = \text{Turbine Release}$$

15.1.1.2 Flow Tables

In this method, you define a table relating the flow to the Turbine Release and the flow to the Min Bypass. You can also specify the Min Bypass which takes precedence over this table.

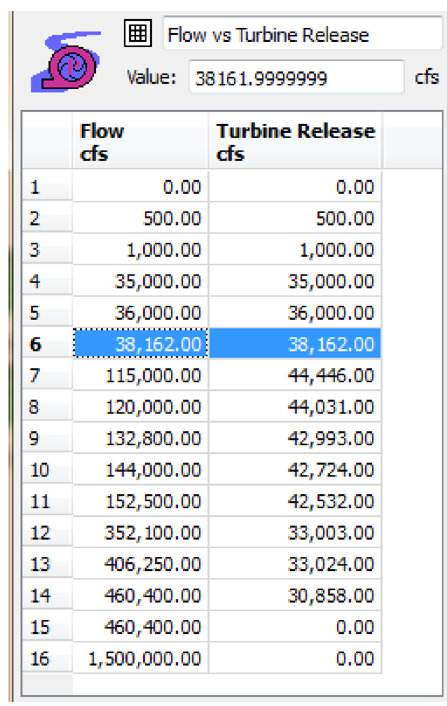
SLOTS SPECIFIC TO THIS METHOD:

☞ **FLOW VS MIN BYPASS**

Type: Table Slot
Units: FLOW VS FLOW
Description: This table relates the flow passing through the Inline Power object to the Min Bypass. The flow used in the first column is the Inflow (or Outflow).
Information: At the beginning of the run, this table is verified to ensure that the Min Bypass value is less than or equal to the Flow value, for each row of the table.
I/O: Required
Links: Not Linkable

☞ **FLOW VS TURBINE RELEASE**

Type: Table Slot
Units: FLOW VS FLOW
Description: This table relates the flow passing through the Inline Power object to the flow passing through the Turbine Release. The flow used in the first column is the Inflow (or Outflow) minus the Min Bypass.
Information: At the beginning of the run, this table is verified to ensure that the Turbine Release value is less than or equal to the Flow value, for each row of the table. A sample table is shown to the right.
I/O: Required
Links: Not Linkable



	Flow cfs	Turbine Release cfs
1	0.00	0.00
2	500.00	500.00
3	1,000.00	1,000.00
4	35,000.00	35,000.00
5	36,000.00	36,000.00
6	38,162.00	38,162.00
7	115,000.00	44,446.00
8	120,000.00	44,031.00
9	132,800.00	42,993.00
10	144,000.00	42,724.00
11	152,500.00	42,532.00
12	352,100.00	33,003.00
13	406,250.00	33,024.00
14	460,400.00	30,858.00
15	460,400.00	0.00
16	1,500,000.00	0.00

METHOD DETAILS:

At the beginning of the run, the Min Turbine Release at Max Generation (*MinTRatMaxGen*) is found on the Flow vs Power Table as follows. The largest power is found on the Flow Power Table and then the corresponding flow is used as the *MinTRatMaxGen*. This value represents the smallest flow that produces the maximum power generation; it is not the maximum turbine release. If the Flow Power Tables is not used, MinTRatMaxGen is set to the Max Turbine Release value.

Then, within the dispatch, this method calculates **Bypass** and **Turbine Release** using the following algorithm:

If the **Turbine Release** value has been input or set by rules, an error will be issued.

Next, the temporary **Flow** variable (this was set by the dispatch method depending on whether Inflow or Outflow is known) is used to interpolate on the **Flow vs Min Bypass** table to obtain a **Min Bypass** value. This value is set on the **Min Bypass** slot, if not input or set by a rule.

A temporary *PowerFlow* variable is computed as: $PowerFlow = Flow - \text{Min Bypass}$

The *PowerFlow* value is used to interpolate on the **Flow vs Turbine Release** table to obtain a *tempTurbineRelease*. Note, *PowerFlow* and *tempTurbineRelease* are not necessarily equal.

If the *PowerFlow* is greater than the *MinTRatMaxGen*,

$$\text{Reduced Max Turbine Release} = tempTurbineRelease \times \text{Power Plant Cap Fraction}$$

If the *PowerFlow* is less than or equal to *MinTRatMaxGen*,

$$\text{Reduced Max Turbine Release} = \text{MinTRatMaxGen} \times \text{Power Plant Cap Fraction}$$

Finally, Turbine Release is set:

$$\text{Turbine Release} = MIN(PowerFlow, \text{Reduced Max Turbine Release})$$

An Error is issued if the Turbine Release is greater than the Max Turbine Release slot value.

Then final Bypass is computed as: $Bypass = Inflow - \text{Turbine Release}$

The *PowerFlow* values is then passed to the Inline Power method as described **HERE (Section 15.1.2.2)**.

15.1.2 Inline Power

These methods determine how power is calculated.

15.1.2.1 None

This default method does no power calculation and has no slots specific to the method.

15.1.2.2 Flow vs. Power Table

This method looks up the power from a table based on the flow rate through the turbines. It uses a linear interpolation to calculate the power. It is important to have enough data values if the power curve is not linear.

Slots Specific to this Method:

☞ Energy

Type: Series Slot
Units: ENERGY
Description: Product of power generated and the timestep length
Information:
I/O: Output only
Links: Usually not linked.

☞ Hydro Capacity

Type: Series Slot
Units: POWER
Description: Maximum power capacity of the inline power plant.
Information: This slot is utilized in optimization
I/O: Output only
Links: Linkable, usually to the same slot on the Thermal object.

☞ Power

Type: Series Slot
Units: POWER
Description: Power generated by flow through the units
Information: Calculated using the PowerFlow and Flow Vs Power Table
I/O: Output only
Links: Not linkable

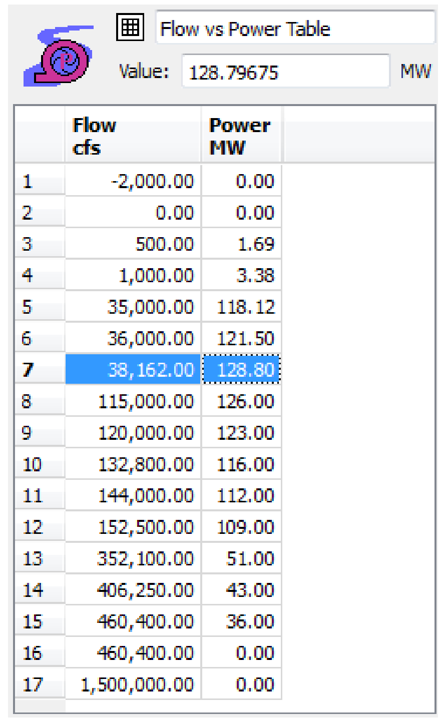
☞ Flow vs Power Table

Type: Table Slot
Units: FLOW VS. POWER
Description: Power produced at that given flow rate.These tables should include any maximum or minimum flow rates through the plant and the corresponding values. In theory, the upstream control should account for these maximums and minimums but having them in the table is more complete.
Information:
I/O: Required Input
Links: Not Linkable

When the “Specify Flows” method, HERE (Section 15.1.1.1), is selected:

The **Power** slot is computed via linear interpolation of the **Flow vs Power Table** using the *PowerFlow* intermediate value as described in this method HERE (Section 15.1.1). **Energy** is then computed as the product of **Power** and the timestep length. The **Hydro Capacity** slot is computed as the **Max Turbine Release** slot value multiplied by the **Power Plant Cap Fraction**.

When the "Flow Tables" method, HERE (Section 15.1.1.2), is selected:



	Flow cfs	Power MW
1	-2,000.00	0.00
2	0.00	0.00
3	500.00	1.69
4	1,000.00	3.38
5	35,000.00	118.12
6	36,000.00	121.50
7	38,162.00	128.80
8	115,000.00	126.00
9	120,000.00	123.00
10	132,800.00	116.00
11	144,000.00	112.00
12	152,500.00	109.00
13	352,100.00	51.00
14	406,250.00	43.00
15	460,400.00	36.00
16	460,400.00	0.00
17	1,500,000.00	0.00

A sample Flow vs Power Table is shown to the right. This method uses the *PowerFlow* intermediate variable and *MinTRatMaxGen* as described HERE (Section 15.1.1.2), The maximum of the *PowerFlow* and the *MinTRatMaxGen* is looked up on **Flow vs Power Table** to get the temp*Capacity*. The temp*Capacity* is then multiplied by the **Power Plant Cap Fraction** and set on the **Hydro Capacity** slot.

The *PowerFlow* is then looked up on the **Flow vs Power Table** to get the *tempHydropower*.

$$\text{Power} = MIN(\text{Hydro Capacity, tempHydropower})$$

Energy is then computed as the product of **Power** and the timestep length.

15.1.2.3 Specify Units Generating

With this method, you specify the turbine capacity and generating capacity for each unit, and for each timestep, you optionally specify the fraction of capacity at which each unit is generating or specify the **Unit Turbine Release**. The method then calculates the **Unit Power** and **Unit Energy** as well as the total plant **Turbine Release**, **Power** and **Energy**. This method cannot be selected in combination with the **Flow Tables** method in the **Inline Turbine Release and Bypass** category.

☞ SLOTS SPECIFIC TO THIS METHOD:

☞ NUMBER OF UNITS

Type: Scalar Slot
Units: NONE
Description: The number of units (generators) in the plant
Information: This must be an integer greater than or equal to 1. If the value is not an integer, RiverWare will round the value down to the nearest integer.
I/O: Required Input
Links: Not linkable

☞ UNIT CAPACITY

Type: Table Slot
Units: FLOW, POWER
Description: The turbine capacity and generating capacity of each unit
Information: The table has a row for each unit. The first column is the unit turbine capacity, the second column is the unit generating capacity. At the start of a run, RiverWare will automatically resize the number of rows to match the value in the Number of Units slot if they do not match already.
I/O: Unit generating capacity (second column) is Required Input. Unit turbine capacity (first column) is optional input.
Links: Not linkable

☞ UNIT GENERATION FRACTION

Type: Agg Series Slot
Units: FRACTION
Description: The slot has one column for each unit and represents the fraction of full capacity at which the unit is generating.
Information: If the corresponding **Unit Turbine Release** is specified (input or rules) and the unit turbine capacity is provided (**Unit Capacity** slot) then the value is calculated as the **Unit Turbine Release** divided by unit turbine capacity. Otherwise, if no value is provided it will default to 1 (generating at full capacity). A value outside the range of 0 to 1 will cause the run to abort with an error message. Note that if this slot is set by rules, the value in the first column (Unit 1) must be set in order to cause the Inline Power object to redispatch with the new **Unit Generation Fraction** value. Setting a value with rules in one of the other columns without setting the first column will not cause the object to redispatch. You cannot specify both **Unit Turbine Release** and **Unit Generation Fraction** for the same unit.
I/O: Input or Output; if not input and if **Unit Turbine Release** is not input, then it defaults to 1
Links: Not linkable

☞ Unit Turbine Release

Type: Agg Series Slot
Units: FLOW
Description: The flow through each turbine, one column for each unit (turbine)
Information: If not specified (input or rules), the value will be calculated as the unit turbine capacity (**Unit Capacity** slot) multiplied by the **Unit Generation Fraction**. If the unit turbine capacity is not provided, the value will be NaN. You cannot specify both **Unit Turbine Release** and **Unit Generation Fraction** for the same unit.
I/O: Input or output
Links: Not linkable

☞ Unit Power

Type: Agg Series Slot
Units: POWER
Description: The power that is generated by each unit, **Unit Capacity** multiplied by **Unit Generation Fraction**
Information: One column for each unit
I/O: Output only
Links: Not linkable

☞ Unit Energy

Type: Agg Series Slot
Units: ENERGY
Description: The energy that is generated by each unit, **Unit Power** multiplied by the timestep length
Information: One column for each unit
I/O: Output only
Links: Not linkable

☞ Power

Type: Series Slot
Units: POWER
Description: The total power generation of the plant, the sum of **Unit Power** for all units
Information:
I/O: Output only
Links: Linkable

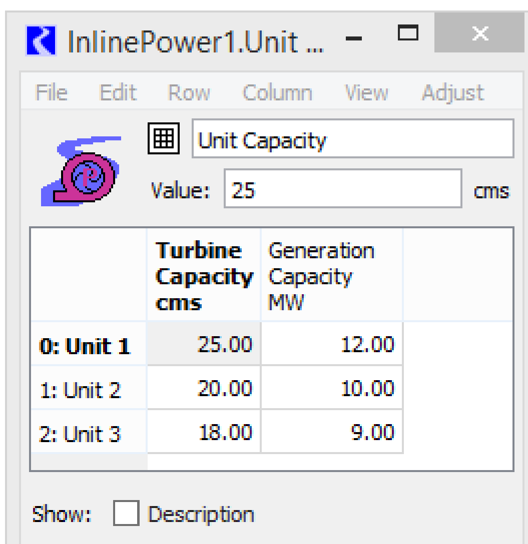
☞ **Energy**

Type: Series Slot
Units: ENERGY
Description: The total energy generation of the plant, **Power** multiplied by the timestep length
Information:
I/O: Output only
Links: Linkable

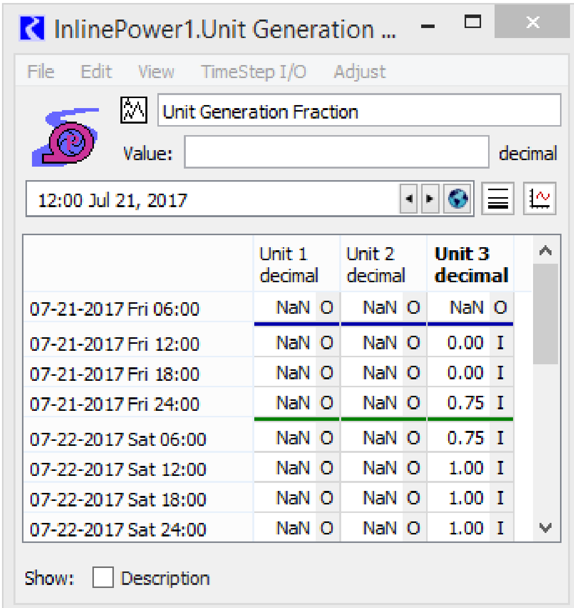
The user must specify the **Number of Units** scalar slot. This must be an integer greater than or equal to 1. If the value is not an integer, RiverWare will round the value down to the nearest integer. At the start of the run, RiverWare will set the number of rows in the **Unit Capacity** slot and number of columns in the **Unit Generation Fraction, Unit Turbine Release, Unit Power** and **Unit Energy** slots to match the value in the **Number of Units** scalar slot. (Note that this could delete rows or columns with data if the **Number of Units** value is reduced.)

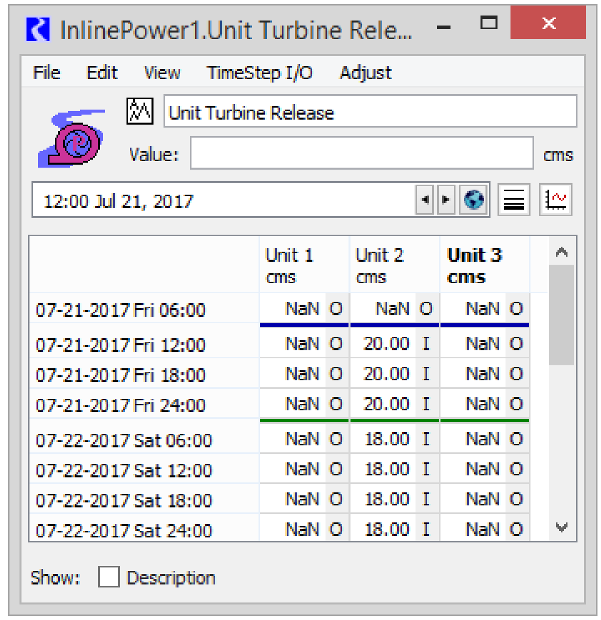
In order to initially add rows and columns to these slots, first set the **Number of Units** scalar slot to the desired value. Then run the model. The run will abort, but the table and agg series slots will be resized to the appropriate number of rows and columns. Then the required input values can be entered in the **Unit Capacity** and **Unit Generation Fraction** or **Unit Turbine Release** slots.

The user must specify the generating capacity value for each unit in the **Unit Capacity** table slot. The turbine capacity values are optional inputs. These are static, physical characteristic parameters. An example is shown below.



The user can optionally provide time series values (input or rules) for each unit in the **Unit Generation Fraction** and **Unit Turbine Release** agg series slots. If values are not provided for **Unit Turbine Release** or **Unit Generation Fraction**, then **Unit Generation Fraction** will default to 1. Specifying both **Unit Generation Fraction** and **Unit Turbine Release** for the same unit is not permitted.





Note: When setting the **Unit Generation Fraction** or **Unit Turbine Release** values by rules, only values set in the "**Unit 1**" column will trigger (re)dispatching of the Inline Power object. Values set in subsequent columns will not cause the object to dispatch.

The method then performs the following computations:

FOR each unit *i*

IF Unit Turbine Release$_i$ is input or set by rules

Use the specified **Unit Turbine Release** to calculate the **Unit Generation Fraction**, but first check against the unit capacity.

IF Unit Turbine Capacity$_i$ is input

IF $\text{Unit Turbine Release}_i > \text{Unit Turbine Capacity}_i$

Abort with and error message.

ELSE

$$\text{Unit Generation Fraction}_i = \frac{\text{Unit Turbine Release}_i}{\text{Unit Turbine Capacity}_i}$$

END IF

ELSE

If there is no unit turbine capacity, then **Unit Generation Fraction** will just get set to 1.

$$\text{Unit Generation Fraction}_i = 1$$

END IF

ELSE (Unit Turbine Release$_i$ is not specified)

IF Unit Turbine Capacity$_i$ is input

Unit Generation Fraction will have been set to 1 if not input or set by rules.

$$\text{Unit Turbine Release}_i = \text{Unit Generation Fraction}_i \times \text{Unit Turbine Capacity}_i$$

ELSE

Unit Turbine Release cannot be calculated so it will remain as NaN.

END IF

END IF

$$\text{Unit Power}_i = \text{Unit Generation Fraction}_i \times \text{Unit Generating Capacity}_i$$

$$\text{Unit Energy}_i = \text{Unit Power}_i \times Timestep$$

END FOR

The **Power** and **Energy** slots are then calculated as the sum of the **Unit Power** and **Unit Energy**.

$$\text{Power} = \sum_{\text{all i}} \text{Unit Power}_i$$

$$\text{Energy} = \sum_{\text{all i}} \text{Unit Energy}_i$$

The the total plant **Turbine Release** can only be calculated by this method if every unit has either the unit turbine capacity (**Unit Capacity** slot) or **Unit Turbine Release** or both specified (input or set by rules). If one or more units has neither of these value specified, then **Turbine Release** and **Bypass** will not get set by this method. The values will come from the selected **Inline Turbine Release and Bypass** method. If **Turbine Release** is set by this method, it will be checked against plant limits. Then **Bypass** will get set as well.

IF Turbine Release can be calculated

$$\text{Turbine Release} = \sum_{\text{all i}} \text{Unit Turbine Release}_i$$

IF $\text{Turbine Release} > \text{Max Turbine Release} \times \text{Power Plant Cap Fraction}$

Abort with an error message

END IF

IF Min Bypass is specified (input or rules)

IF Turbine Release > Outflow – Min Bypass

Abort with an error message

END IF

ELSE

IF Turbine Release > Outflow

Abort with an error message

END IF

END IF

Bypass = Outflow – Turbine Release

ELSE (Total plant Turbine Release cannot be calculated)

Leave Turbine Release and Bypass as set by the Inline Power Turbine Release and Bypass method.

END IF

15.1.3 Inline Power Solution Direction

These methods limit the available dispatch methods.

15.1.3.1 Solve Upstream or Downstream

This default method allows the inline power object to solve in either direction based on the knowns and unknowns. Both the **Solve Given Inflow** and **Solve Given Outflow** dispatch methods are available.

15.1.3.2 Solve Downstream Only

This method limits the inline power object's available dispatch methods to **Solve Given Inflow**. This method may be useful if rule priorities force the inline power object to dispatch with the undesired method resulting in excessive iterations.

15.1.4 Inline Turbine Ramping

These methods model turbine ramping.

15.1.4.1 None

This default method assumes that either ramping is not taking place, or that the user does not care to account for the cost of ramping. No slots are associated with this method.

15.1.4.2 Track Ramping

This method models the cost of ramping. The maintenance cost from wear and tear due to ramping will vary depending on the condition of the power plant. Ramping cost is set equal to a user defined Unit Ramping cost times the sum of Turbine Increase and Turbine Decrease.

$$RampingCost = UnitRampingCost \times (TurbineIncrease + TurbineDecrease)$$

Slots Specific to this Method:

☞ **RAMPING COST**

Type: Series Slot
Units: VALUE ($)
Description: Loss of money per timestep due to ramping.
Information: Set equal to the sum of Unit Ramping Cost times the sum of Turbine Increase and Turbine Decrease.
I/O: Output
Links: Linkable, usually to the same slot on the Thermal object.

☞ **TURBINE DECREASE**

Type: Series Slot
Units: FLOW
Description: Decrease in the turbine release relative to last timestep.
Information: Set to the absolute value of Turbine Increase when Turbine Increase is negative, otherwise set to zero.
I/O: Output
Links: Linkable, usually to the same slot on the Thermal object

☞ **TURBINE INCREASE**

Type: Series Slot
Units: FLOW
Description: Increase in turbine release relative to last timestep.
Information: If turbine increase is negative, the slot is set to zero.
I/O: Output
Links: Linkable, usually to the same slot on the Thermal object.

☞ **UNIT RAMPING COST**

Type: Table Slot
Units: $PER FLOW
Description: Maintenance cost of wear and tear due to ramping.
Information: A single value in a table slot determined by users depending on the condition of the power plant.
I/O: Input
Links: Not Linkable

15.1.5 Spill Cost Category

These methods calculate the cost associated with spill.

15.1.5.1 None

This default method does no spill cost calculations. No slots are associated with this method.

15.1.5.2 Unit Spill Cost

The Unit Spill Cost method is used to calculated the money that is lost when water goes to the bypass as spill rather than through the turbines. Spill cost is calculated as the local variable spill times the Unit Spill Cost.

$$SpillCost = spill \times UnitSpillCost$$

SLOTS SPECIFIC TO THIS METHOD:

☞ **SPILL COST**

Type: Series Slot
Units: VALUE ($)
Description: Loss of money per timestep due to spill
Information: Set equal to the unit spill cost times the spill.
I/O: Output
Links: Linkable, usually to the same slot on the Thermal object.

☞ **UNIT SPILL COST**

Type: Table Slot
Units: VALUEPERFLOW
Description: Money from hydropower lost due to spill.
Information: A single value in a table slot determined by users based on the market value for hydropower and the power produced per unit flow at the power plant.
I/O: Input
Links: Not Linkable

15.2 Dispatch Methods

This object has two dispatch methods: **Solve Given Inflow** and **Solve Given Outflow**.

15.2.1 Solve Given Inflow

The required knowns and unknowns (at the current timestep) are listed below.

REQUIRED KNOWNS

☞ **INFLOW**

REQUIRED UNKNOWNS

☞ **OUTFLOW**

The **Solve Given Inflow** dispatch method solves for Outflow, Bypass, and Turbine Release (if it is not input) based on Inflow.

First, the method sets the temporary Flow variable equal to Inflow. Then, the selected method in the **Inline Turbine Release and Bypass** category, **HERE (Section 15.1.1)**, is executed. This method sets Turbine Release and Bypass

Outflow is then set to the sum of Turbine Release and Bypass:

$$Outflow = TurbineRelease + Bypass$$

A check is performed to ensure that Outflow equals Inflow.

Power and Energy are then calculated with a call to the selected Inline Power method using the computed *PowerFlow* value.

A call to the Ramping methods executes any calculations associated with the selected Ramping method.

Lastly, a call to the Spill Cost methods execute the selected Spill Cost method.

15.2.2 Solve Given Outflow

The required knowns and unknowns (at the current timestep) are listed below.

REQUIRED KNOWNS

☞ **OUTFLOW**

REQUIRED UNKNOWNS

☞ INFLOW

The **Solve Given Outflow** dispatch method solves for Inflow, Bypass, and Turbine Release (if it is not input) based on Inflow.

First, the method sets the temporary Flow variable equal to Outflow. Then, the selected method in the **Inline Turbine Release and Bypass** category, HERE (Section 15.1.1), is executed. This method sets Turbine Release and Bypass

Next, the Inflow is then set to the sum of Turbine Release and Bypass:

$$Inflow = TurbineRelease + Bypass$$

A check is performed to ensure that Outflow equals Inflow.

Power and Energy are then calculated with a call to the selected Inline Power method using the computed *PowerFlow* value.

A call to the Ramping methods executes any calculations associated with the selected Ramping method.

Lastly, a call to the Spill Cost methods execute the selected Spill Cost method.

16. Inline Pump

This object models a booster pump station. It contains methods that control the solution direction, calculate the head added, and calculate the power consumed.

General Slots

☞ Inflow

Type: SeriesSlot

Units: Flow

Description: Flow at the entrance of the object

I/O: Input, Output, or set through a link

Links: Usually linked to the outflow slot on another object

☞ Outflow

Type: SeriesSlot

Units: Flow

Description: Flow at the exit of the object

I/O: Input, Output, or set through a link

Links: Usually linked to the inflow slot on another object.

16.1 User Methods

16.1.1 Inline Pump Solution Direction

The user selectable methods for Inline Pump Solution Direction will include “Solve Upstream or Downstream” and “Solve Downstream Only”. “Solve Upstream or Downstream” is the default method and should remain selected for basic simulation. “Solve Downstream Only” may need to be selected for Rulebased Simulation if the Inline Pump will only solve in the downstream direction. When this method is selected, only the solveOutflowGivenInflow dispatch method is available.

16.1.1.1 Solve Upstream or Downstream

This is the default method and should remain selected for basic simulation. There are no slots or calculations specifically associated with this method. It simply allows all dispatch methods to be active.

16.1.1.2 Solve Downstream Only

This method may need to be selected in Rulebased Simulations if the Inline Pump solves in the downstream direction only. When this method is selected, only the solveOutflowGivenInflow dispatch method is available.

16.1.2 Inline Pump Hydraulics

The user selectable methods for Inline Pump Hydraulics includes “None” and “Head Lookup”. “None” is the default method and makes no other calculations. The “Head Lookup” method uses the Pump Curve table and the inflow to calculate head added by the pump.

16.1.2.1 None

This is the default method and performs no calculations.

16.1.2.2 Head Lookup

This method will use the Pump Curve table and the flow rate to calculate the head added by the pump. In this method, either Inflow Head or Outflow Head must be given (either set as an input or set through a link) in order to calculate the other. If Inflow Head is input or set through a link, Outflow Head is calculated. If Outflow Head is input or set through a link, Inflow Head is calculated.

Slots Specific To This Method

☞ Inflow Head

Type: SeriesSlot

Units: Length

Description: Inflow Head at the Inline Pump Object

I/O: Input or Output

Links: Usually linked to the outflow head slot on another object

☞ Outflow Head

Type: SeriesSlot

Units: Length

Description: Outflow Head at the Inline Pump Object

I/O: Input or Output

Links: Usually linked to the inflow head slot on another object

☞ Head Added

Type: SeriesSlot

Units: Length

Description: Operating head added by the Inline Pump

I/O: Output

Links: not linkable

☞ Minor Losses

Type: ScalarSlot

Units: LENGTH

Description: Minor Losses at the Inline Pump

I/O: Input Only

Links: not linkable

☞ Pump Curve Table

Type: TableSlot

Units: LENGTH VS. FLOW

Description: A table that exhibits a curve for Head versus Flow. It is used to interpolate head added by the pump from the flow rate

I/O: Input Only

Links: not linkable

☞ Pump Status

Type: SeriesSlot

Units: NOUNITS

Description: This slot defines the status of the pump at each timestep. A value of 1 indicates the pump is on, a value of 0 indicates the pump is off. Unless input, the pump is assumed to be on.

I/O: Input or set by a rule; defaults to 1

Links: Usually not linked

The calculations used for this method proceeds as follows. The Pump Status value is checked to determine whether the pump is on or off. If the pump is off, the HeadAdded is set to zero. If the pump is on (=1), the Inflow is used as the lookup value in the Pump Curve Table to determine the Head Added by the pump. The Outflow Head is then calculated as:

$$OutflowHead = InflowHead + HeadAdded - MinorLosses$$

If solving given Outflow, the Inflow Head is calculated as:

$$InflowHead = OutflowHead - HeadAdded + MinorLosses$$

16.1.3 Inline Pump Energy

The Inline Pump Energy category is used specify the methods that determine the power consumed by the Inline Pump. The category is dependent on having the Head Lookup method selected. There are two methods: “None” and “Energy Equation”.

16.1.3.1 None

This is the default method. It performs no calculations.

16.1.3.2 Energy Equation

The Energy Equation method uses the standard formula to determine the power consumed by the pump.

Slots Specific To This Method

☞ **Power Consumed**

Type: SeriesSlot
Units: POWER
Description: Power consumed by Inline Pump
I/O: Output Only
Links: not linkable

☞ **Energy Consumed**

Type: SeriesSlot
Units: ENERGY
Description: Energy consumed by Inline Pump
I/O: Output Only
Links: not linkable

☞ **Density of Water**

Type: Scalar Slot
Units: DENSITY
Description: Density of the Water
I/O: Input Only
Links: not linkable

☞ Pump Efficiency

Type: ScalarSlot
Units: DECIMAL
Description: Efficiency of the Inline Pump
I/O: Input Only
Links: not linkable

The energy and power consumed by the pump are calculated according to the following formulas.

$$PowerConsumed = \frac{(Inflow \times DensityofWater \times Gravity \times HeadAdded)}{1000000 \times PumpEfficiency}$$

$$EnergyConsumed = PowerConsumed \times TimestepLength$$

16.2 Dispatch Methods

This object solves for inflow or outflow depending on the known and unknown.

16.2.1 solveInflowGivenOutflow

The required knowns and unknowns (at the current timestep) are listed below.

REQUIRED KNOWN

☞ **OUTFLOW**

REQUIRED UNKNOWN

☞ **INFLOW**

If an Outflow Head is linked, the dispatch method will wait for a value in this slot before solving. Then the Inflow is solved by:

$$Inflow = Outflow$$

16.2.2 solveOutflowGivenInflow

The required knowns and unknowns (at the current timestep) are listed below.

REQUIRED KNOWN

☞ **INFLOW**

REQUIRED UNKNOWN

☞ **OUTFLOW**

If an Inflow Head is linked, the dispatch method will wait for a value in this slot before solving. This dispatch method solves for Outflow as:

$$Outflow = Inflow$$

17. Level Power Reservoir

Similar to a Storage Reservoir, except it is able to model power production facilities on the reservoir. This object assumes the reservoir water surface is level.

General Slots

☛ Canal Flow

Type: Series Slot
Units: FLOW
Description: flow into (out of) the reservoir from (to) a canal
Information: May be linked to either the Flow 1 or Flow 2 slot of the Canal object. If not linked, the slot is set to zero.
I/O: Output only
Links: May be linked to either the Flow 1 or Flow 2 slot of the Canal object. If not linked, the slot is set to zero.

☛ Convergence Percentage

Type: Table
Units: NONE
Description: A percentage value ranging from 0 to 1 used for convergence in all iterative calculations
Information: Click HERE (Appendix A: Reservoir Convergence) for more information on the convergence algorithm
I/O: Optional; defaults to 0.0001 if not input.
Links: Not linkable

☛ Diversion

Type: Series
Units: FLOW
Description: flow from the reservoir to a diverting object
Information: If not linked or input it is set to zero.
I/O: Optional; may be input or linked or neither
Links: May be linked to the Total Diversion slot on an Agg Diversion Site or the Total delivery Request slot on an AggDistribution Canal.

☛ Diversion Capacity

Type: Scalar Slot
Units: FLOW
Description: used to hold the maximum diversion physically possible from the reservoir
Information: This slot is used in the accounting system for allocation purposes and can be used in Rulebased Simulation
I/O: Input only
Links: Not linkable

☛ Elevation Volume Table

Type: Table
Units: LENGTH vs. VOLUME
Description: Reservoir Pool Elevation vs. Reservoir Storage
Information:
I/O: Required input
Links: Not linkable

☛ Energy

Type: Series Slot
Units: ENERGY
Description: product of the power generated by flow through the turbines and the length of the timestep
Information: This slot may also take the BEST_EFFICIENCY or MAX_CAPACITY flags, which allow Energy to act as input for dispatching, but solve for the value of energy assuming the generators are operating at best efficiency or maximum capacity.
I/O: Optional; if not input by the user, Energy is computed in the power calculations.
Links: Usually not linked

☛ Flow FROM Pumped Storage

Type: Series Slot
Units: FLOW
Description: flow into the reservoir from a pumped storage reservoir
Information: May be linked to the Outflow slot of a Pumped Storage object.
I/O: Optional; usually linked if used.
Links: May be linked to the Outflow slot of a Pumped Storage object.

☛ Flow TO Pumped Storage

Type: Series Slot
Units: FLOW
Description: flow out of the reservoir into a pumped storage reservoir

Information:
I/O: Optional; usually linked if used
Links: May be linked to the Pumped Flow slot of a Pumped Storage object.

☛ INFLOW

Type: Multi Slot
Units: FLOW
Description: inflow into the reservoir from upstream
Information:
I/O: Optional; if not input by the user, it is set through either mass balance computations or the propagation of values across the link.
Links: May be linked to one or more outflow slots of upstream objects.

☛ MAX ITERATIONS

Type: Table
Units: NOUNITS
Description: maximum number of allowable iterations for iterative loops in the solution algorithms
Information: Used in conjunction with Convergence Percentage as a stopping criterion for iterative calculations.
I/O: Optional; defaults to 100 if not input.
Links: Not linkable

☛ OPERATING HEAD

Type: Series Slot
Units: LENGTH
Description: elevation difference between the average Pool Elevation and the average Tailwater Elevation during a timestep
Information:
I/O: Output only
Links: Usually not linked

☛ OUTFLOW

Type: Series Slot
Units: FLOW
Description: outflow from reservoir
Information: The outflow from a Level Power Reservoir is equal to the sum of the Turbine Release and the Spill. May be linked to the inflow slot of a downstream object. If not input by the user, it is set through either the mass balance computations or the propagation of values across the link.
I/O: Optional; if not input by the user, it is set through either the mass balance computations or the propagation of values across the link.

Links: May be linked to the inflow slot of a downstream object.

☛ Pool Elevation

Type: Series Slot
Units: LENGTH
Description: elevation of the water surface of the Reservoir
Information: There must be an initial value for either Storage or Pool Elevation given by the user for the first timestep.
I/O: Optional; if not input by the user, it is solved by the mass balance computations. It may take a TARGET flag indicated by the user for target operation solution.
Links: May be linked to Tailwater Elevation or Tailwater Base Value of an upstream object or to Elevation 1 or Elevation 2 of a Canal object.

☛ Power

Type: Series Slot
Units: POWER
Description: power generated by flow through the turbines
Information: Calculated by the power methods and cannot be input by the user.
I/O: Output only
Links: Not linkable

☛ Return Flow

Type: Multi Slot
Units: FLOW
Description: flow returning from a diverting object
Information:
I/O: Optional; defaults to zero if not linked or input.
Links: May be linked to one or more Return Flow slots on Water User objects or the Total Return Flow slot on the Agg Diversion Site objects.

☛ Spill

Type: Series Slot
Units: FLOW
Description: sum of the regulated and unregulated spills and bypass
Information: May be input or solved for by **RiverWare**™(see spill calculation methods).
I/O: Optional; may be input or solved for by **RiverWare**™ (see spill calculation methods).
Links: Usually not linked

☛ Storage

Type: Series Slot
Units: VOLUME
Description: volume of water stored in the reservoir
Information: May be flagged as a TARGET Storage value by the user. There must be an initial value for either Storage or Pool Elevation given by the user for the first timestep. If flagged as a TARGET, a target operation solution is used.
I/O: Optional; if not input by the user, it is set through mass balance computations.
Links: Usually not linked

☛ Tailwater Elevation

Type: Series
Units: LENGTH
Description: water surface elevation on the downstream side of the dam
Information: It can be linked to the Pool Elevation or Backwater Elevation of a downstream reservoir if the “Linked or Input” method is selected for the Tailwater category. Otherwise, it is calculated by the user method selected. It is used to compute the Operating Head used in the power calculations.
I/O: Optional; can be input, linked or calculated.
Links: It can be linked to the Pool Elevation or Backwater Elevation of a downstream reservoir if the “Linked or Input” method is selected for the Tailwater category.

☛ Total Inflows

Type: Series
Units: FLOW
Description: Summary slot displaying the flows into and out of the reservoir excluding the flows through the outlet works
Information: Total Inflows is calculated using the following equation:

$$\text{Total Inflows} = \text{Inflow} + \text{Canal Flow} + \text{Hydrologic Inflow} + \text{Hydrologic Inflow Adjust} + \text{Hydrologic Inflow Forecast} + \text{Return Flow} + \text{Flow FROM Pumped Storage} - \text{Flow TO Pumped Storage} - \text{Diversion}$$

Any component that is not in use or is not valid defaults to zero.
I/O: Output only
Links: Not linkable

☛ Inflow Sum

Type: Series
Units: FLOW
Description: Sum of the total flows entering the reservoir at each timestep

Information: Inflow Sum is calculated using the following equation:

Inflow Sum = Inflow + Canal Flow +
Hydrologic Inflow + Hydrologic Inflow Adjust + Hydrologic Inflow Forecast +
Return Flow + Flow FROM Pumped Storage

I/O: Output only
Links: Not Linkable

☛ TURBINE RELEASE

Type: Series Slot
Units: FLOW
Description: flow through the turbines of a power reservoir (excluding spill)
Information:
I/O: Optional; solved for if not input.
Links: Usually not linked

17.1 User Methods

17.1.1 Power

The Power category user methods calculate the flow through the turbines (Turbine Release) and the Power and Energy generated. These methods require that the total Outflow of the Reservoir be known

17.1.1.1 None

This is the default method in the Power category. It contains no calculations for Power or Energy. There are no slots specifically associated with this method.

Slots Specific to this Method

☛ None

The method first checks that Energy and Turbine Release are not input by the user. These slots cannot be input when None is the selected method. If either of these two slots are input, a **RiverWare**™ error will be posted and the simulation run aborted.

Next, the selected method in the Power Plant Failure category is executed. This sets the Power Plant Cap Fraction if necessary and checks for plant shutoff/failure. If the plant is shutoff/failed, the turbine Release is set to 0.0.

Otherwise, the Turbine Release is calculated as the difference between Outflow and Spill.

17.1.1.2 No Power Turbine Flow

The No Power Turbine Flow method is used to model Turbine Release without any power generation. Turbine Release is calculated as the Outflow minus Spill. The computed Turbine Release can not be larger than the Max Flow Through Turbines.

Slots Specific to this Method

☛ Max Flow Through Turbines

Type: Table Slot
Units: LENGTH vs. FLOW
Description: relationship between Pool Elevation and Turbine Capacity
Information:
I/O: Required input
Links: Not linkable

The method first checks that Energy and Turbine Release are not input by the user. These slots are not valid for user input when the No Power Turbine Flow method is selected. If either of these two slots is input, a **RiverWare™** error will be posted and the simulation run will be aborted. Pool Elevation is then used in an interpolation scheme to determine the maximum release from the Max Turbine Flow table.

Next, the selected method in the Power Plant Failure category is executed. This sets the Power Plant Cap Fraction if necessary and checks for plant shutoff/failure. If the plant is shutoff/failed, the turbine Release is set to 0.0.

Otherwise, the Turbine Release is set as either Outflow minus Spill or maximum release. It is set as the lesser of the two values because the Turbine Release must be less than the Turbine Capacity.

17.1.1.3 Plant Power Coefficient

The Plant Power Coefficient method calculates the Power and Energy generated based on the whole plant characteristics. If the Power Coefficient is specified, the Power is calculated directly, unless the **BEST EFFICIENCY** or **MAX CAPACITY** flag is set on Energy. If its not input, the Power Coefficient is found from the interpolation of the Best or Max Turbine Q and Power Coefficient tables using the current Operating Head. If the Turbine Release is less than the Best Turbine Q, the Best Power Coefficient Table is used. If the Turbine Release is greater than the Max Turbine Q, then the Max Power Coefficient Table is used. If the Turbine Release is between the two, an intermediate Power Coefficient Value is found by interpolation.

Slots Specific to this Method

☛ Best Hydro Capacity

Type: Series Slot
Units: POWER
Description: most efficient hydro capacity of the plant at the current timestep
Information: Solved iteratively based on Best Turbine Q and the Best Power Coefficient.
I/O: Output only
Links: Not linkable

☛ Best Power Coefficient

Type: Table Slot
Units: LENGTH vs. POWER per FLOW
Description: Operating Head vs. most efficient power coefficient
Information: The Power Coefficient relates turbine release to power generated. The Best Power Coefficient represents the most efficient power generation.
I/O: Required input
Links: Not linkable

☛ Best Turbine Q

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Operating Head vs. flow through turbine for most efficient power generation
Information:
I/O: Required input
Links: Not linkable

☛ Hydro Capacity

Type: Agg Series Slot
Units: POWER
Description: maximum hydro capacity of plant at the current timestep
Information: Solved iteratively based on Max Turbine Q and the Maximum Power Coefficient.
I/O: Output only
Links: Usually not linked

☛ Max Power Coefficient

Type: Table Slot
Units: LENGTH vs. POWER per FLOW
Description: Operating Head vs. maximum power coefficient
Information: The Power Coefficient relates turbine release to power generated. The Max Power Coefficient represents the maximum Turbine Release.
I/O: Required input
Links: Not linkable

☛ Max Turbine Q

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Operating Head vs. maximum flow through the turbine
Information:
I/O: Required input
Links: Not linkable

☛ Minimum Power Elevation

Type: Table Slot
Units: LENGTH
Description: minimum Pool Elevation for power production
Information:
I/O: Required input
Links: Not linkable

☛ Plant Power Limit

Type: Series Slot
Units: POWER
Description: Power output is limited to this value
Information:
I/O: Optional; This constraint on power is only applied if the user inputs a value for the timestep.
Links: Not linkable

☛ Power Coefficient

Type: Series Slot
Units: POWER PER FLOW
Description: power generated per unit power release
Information:
I/O: Optional; can be input or calculated.
Links: Usually not linked

☛ Power Plant Cap Fraction

Type: Series Slot
Units: FRACTION
Description: the percentage of full capacity of the turbine units in the hydropower plant
Information:
I/O: Optional; The value of this slot defaults to 100% if not input by user.
Links: Not linkable

This method performs calculations to compute the power generated at each timestep.

First, Tailwater Elevation and Operating Head are determined based on the user method selected in the Tailwater category.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the Minimum Power Elevation was input by the user. If no value was input, a **RiverWare**™ error is posted and the simulation run is aborted. If the previous Pool Elevation is less than the Minimum Power Elevation or the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

4. Energy, Power, Hydro Capacity, and Best Hydro Capacity are set equal to zero.

5. If the Turbine Release is input or already set from the Dispatch Method “solveMB_givenInflowRelease,” a **RiverWare**™ error is flagged and the run is aborted.

6. Turbine Release is set equal to zero.

Now, Operating Head is used to determine the maximum power release through interpolation on the Max Turbine Q table. The maximum power release value is multiplied by the Power Plant Cap Fraction to account for the state of the turbine units.

If Turbine Release is already set from the dispatch method “solveMB_givenInflowRelease”, the following checks are performed:

- If Turbine Release is greater than Outflow - Spill, a **RiverWare**™ error is posted reading, “Requested Power Release is Greater than Outflow - Spill” and the run is aborted.
- If Turbine Release is greater than the maximum power release, a **RiverWare**™ error is posted reading, “Requested Turbine Release is greater than Maximum Turbine Capacity” and the run is aborted.

If the Turbine Release was input by the user, a **RiverWare**™ error is posted and the run is aborted. If neither the Energy nor the Turbine Release were input and the Energy was not set by a rules, the Turbine Release is set equal to the lesser of the Maximum Power Release or the value of the Outflow minus the Spill.

Using the calculated value of Operating Head, QmaxTemp and QbestTemp are obtained from the Max Turbine Q Table and the Best Turbine Q Table, respectively. Both values are then multiplied by the Power Plant Cap Fraction to obtain Qmax and Qbest. The Operating Head is also used to determine both the best power coefficient and the max power coefficient through interpolation of the Best Power Coefficient and Max Power Coefficient tables, respectively.

The following calculations are not performed if Energy is Input, set by a Rule, or flagged **BEST EFFICIENCY** or **MAX CAPACITY**. In these cases the Power, Energy, and Power Coefficient have already been calculated in Plant Power Coefficient Release.

If the Power Coefficient is not input by the user, the following steps are performed:

1. If the maximum power coefficient is greater than the best power coefficient, the following **RiverWare**™ error is posted, “best Power Coeff < full gate Power Coeff” and the simulation run is aborted.
2. If Qbest is greater than Qmax, the following **RiverWare**™ error is posted, “Best Turbine Q > Max Turbine Q” and the simulation run is aborted.
3. If Qbest equals Qmax, the Power Coefficient is set equal to the best power coefficient.
4. If none of the previous three conditions are satisfied and the Turbine Release is less than or equal to Qbest, the Power Coefficient is set equal to the best power coefficient.

5. If none of the previous four conditions are satisfied and the Turbine Release is less than Qmax, the Power Coefficient is calculated using the following equation:

$$\text{Power Coefficient} = \text{best power coefficient} + \frac{(\text{Turbine Release} - \text{Qbest})}{(\text{Qmax} - \text{Qbest})} \times (\text{max power coefficient} - \text{best power coefficient})$$

6. If none of the previous four conditions are true, the Power Coefficient is set equal to the max power coefficient.

Power is then calculated using the following equation:

$$\text{Power} = \text{Power Coefficient} \times \text{Turbine Release}$$

If the user has input the Plant Power Limit, the following steps are taken:

1. If the Power Coefficient is input by the user, Power and Turbine Release may need to be recalculated. If the Power is greater than the Plant Power Limit, Power is set equal to the Plant Power Limit and Turbine Release is recalculated as the Plant Power Limit divided by the Power Coefficient.

2. If the Power Coefficient is not input and the Plant Power Limit is exceeded; the Turbine Release, Power, and Power Coefficient may need to be recalculated. If the Power Coefficient is equal to the best power coefficient, the plant is already operating at best efficiency. Therefore, the Turbine Release is set equal to the Plant Power Limit divided by the Power Coefficient and the rest of the flow is spilled. The Power and Power Coefficient do not need to be recalculated.

If the Power Coefficient is not equal to the best power coefficient, Turbine Release, Power, and the Power Coefficient need to be recalculated. This is done through the following steps:

- Temporary variables are calculated from the following equations:

$$\text{power at best} = \text{best power coefficient} \times \text{Qbest}$$

$$\text{power at max} = \text{max power coefficient} \times \text{Qmax}$$

$$\text{Qlimit} = \text{Qbest} + \frac{\text{Plant Power Limit} - \text{power at best}}{(\text{power at max} - \text{power at best}) \times (\text{Qmax} - \text{Qbest})}$$

$$\text{PClimit} = \text{best power coefficient} + \frac{\text{Plant Power Limit} - \text{power at best}}{(\text{power at max} - \text{power at best}) \times (\text{max power coefficient} - \text{best power coefficient})}$$

- If Qlimit is greater than Qmax: the Power Coefficient is set equal to the max power coefficient, Turbine release is set equal to Qmax, and Power is set equal to the power at max.
- If Qlimit is less than Qbest: the Power Coefficient is set equal to the best power coefficient, Turbine Release is set equal to Qlimit, and Power is set equal to the Plant Power Limit.
- If Qlimit is less than Qmax but greater than Qbest: the Power Coefficient is set to equal to PClimit, Turbine Release is set equal to Qlimit, and Power is set equal to the Plant Power Limit.

If the Spilled Energy Power Coefficient is visible on the reservoir and it is not input, it is set equal to the Power Coefficient.

Energy is calculated as Power multiplied by the timestep (in hours).

The following calculations take always take place, regardless of the flag on Energy:

If either the Turbine Release is equal to the maximum power release or the Energy is at the maximum capacity, Hydro Capacity is set equal to Power and the Best Hydro Capacity is obtained from the getHydroCap function. If the energy is at the **Best Efficiency**, the Best Hydro Capacity is set equal to the Power and the Hydro Capacity is obtained from the getHydroCapacity function. If neither the Turbine Release is equal to the maximum power release, the Energy is at the maximum capacity, nor the Energy is at the **Best Efficiency**, both Hydro Capacity and Best Hydro capacity are obtained with the getHydroCap function.

17.1.1.4 Plant Efficiency Curve

The Plant Efficiency Curve method calculates the Power and Energy generated based on the whole plant characteristics. If the Power Coefficient is specified, the Power is calculated directly, unless the **BEST EFFICIENCY** or **MAX CAPACITY** flag is set on Energy. If the Power Coefficient is not input, the Power is found by a 3-D interpolation of the Plant Power Table using the current, average Operating Head and Turbine Release. The Power Coefficient is calculated as Power divided by Turbine Release.

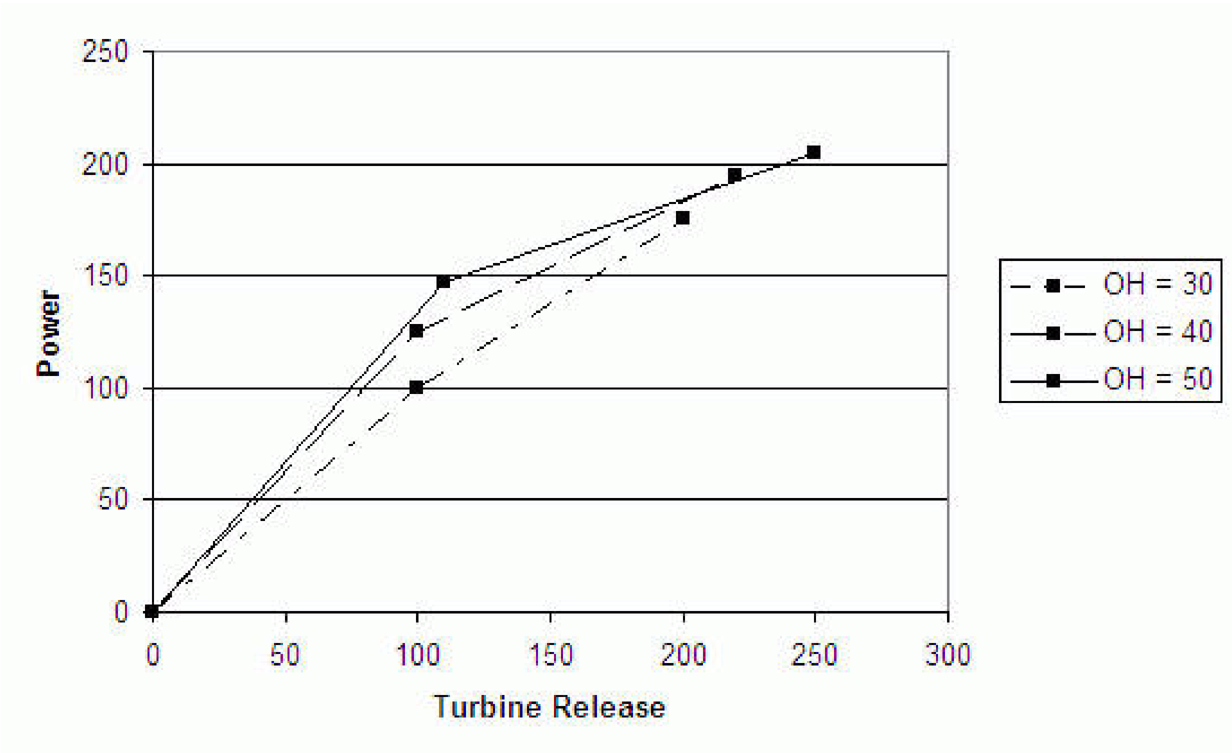
Slots Specific to this Method

☛ Plant Power Table

Type: Table Slot
Units: LENGTH, FLOW, POWER
Description: 3-D table used to determine power using interpolation
Information: Data must be entered into the table in increasing, concave blocks of the same Operating Head for the 3-dimensional table interpolation to work correctly. For every block of the same Operating Head in column 1, Turbine Release should be listed in increasing, concave order in column 2, and the

corresponding Power in column 3. There should also be a point of zero Turbine Release and zero Power for each operating head. The second to last row for each operating head is the point of best efficiency. The last row for each operating head is the point of maximum Turbine Release and maximum Power production. If there are only two rows for a given operating head, both the **best efficiency** and **max capacity** are equal to the second row. The table shown below is an example of the proper way to formulate the Plant Power Table. The graph displays the increasing concave blocks at each operating head.

Operating Head	Turbine Release	Power
30	0	0
30	100	100
30	200	175
40	0	0
40	100	125
40	220	195
50	0	0
50	110	147
50	250	205



I/O: Input Only
Links: Not Linkable

☛ POWER COEFFICIENT

Type: Series Slot
Units: POWER PER FLOW
Description: power generated per unit flow release
I/O: Optional; if input, it is used to compute power. Otherwise, power is computed from the Plant Power Table
Links: Not usually linked

☛ HYDRO CAPACITY

Type: Agg Series Slot
Units: POWER
Description: This is the maximum power that can be produced at the current timestep
Information: Solved for iteratively based on the Operating Head and maximum possible release.
I/O: Output Only

Links: Not Linkable

☛ Best Hydro Capacity

Type: Series Slot
Units: POWER
Description: This is the power that would be produced at the most efficient operating point at the current timestep.
Information: Solved for iteratively based on the most efficient operating point and the corresponding release.
I/O: Output Only
Links: Not Linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: FRACTION
Description: Must be a number less than or equal to 1. If not input, automatically set to 1.
Information: This is the percentage of full capacity of the turbine units in the hydropower plant. For example, if only half of the turbines are operational (and they are all the same), this value would be 0.5
I/O: Can be input by user. If not, value is set to 1.
Links: Not Linkable

☛ Minimum Power Elevation

Type: Series Slot
Units: LENGTH
Description: The minimum elevation at which the reservoir can still produce power.
I/O: Input Only
Links: Not Linkable

☛ Plant Power Limit

Type: Series Slot
Units: POWER
Description: The max power that the plant can produce at a given timestep.
I/O: Optional, only applies if input by user
Links: Not Linkable

☛ Power Curvature Tolerance

Type: Scalar
Units: NONE
Description: The power curvature tolerance is used to account for anomalies in Plant Power Table data and round off error while calculating slopes.

Information: Although the units for the slot are “None”, the comparison is implicitly using (MW/cms).
I/O: Input or defaults to $1X10^{-6}$
Links: Not linkable

At the start of an optimization run, the Plant Power Table is checked for concavity. The slope of each segment for each block is calculated as:

$$Slope_{Segment} = \frac{Power_i - Power_{i-1}}{TurbineRelease_i - TurbineRelease_{i-1}}$$

The table is considered concave if:

$$Slope_{Segment} \leq Slope_{PreviousSegment} + PowerCurvatureTolerance$$

The table is required to be concave for optimization runs, but not for simulation or rulebased simulation runs.

This method performs calculations to compute the power generated at each timestep.

First, Tailwater Elevation and Operating Head are determined based on the user method selected in the Tailwater category.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the Minimum Power Elevation was input by the user. If no value was input, a **RiverWare**™ error is posted and the simulation run is aborted. If the previous Pool Elevation is less than the Minimum Power Elevation or the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Energy, Power, Hydro Capacity, and Best Hydro Capacity are set equal to zero.
2. If the Turbine Release is input or already set from the Dispatch Method “solveMB_givenInflowRelease,” a **RiverWare**™ error is flagged and the run is aborted.
3. Turbine Release is set equal to zero.

Operating Head is used to determine the maximum power release through interpolation on the Plant Power Table. The maximum power release value is multiplied by the Power Plant Cap Fraction to account for the state of the turbine units.

If Turbine Release is already set from the dispatch method "solveMB_givenInflowRelease", the following checks are performed:

- If Turbine Release is greater than Outflow - Spill, a **RiverWare**™ error is posted reading, "Requested Power Release is Greater than Outflow - Spill" and the run is aborted.
- If Turbine Release is greater than the maximum power release, a **RiverWare**™ error is posted reading, "Requested Turbine Release is greater than Maximum Turbine Capacity" and the run is aborted.

If the Turbine Release was input by the user, a **RiverWare**™ error is posted and the run is aborted. If neither the Energy nor the Turbine Release were input and the Energy was not set by a rules, the Turbine Release is set equal to the lesser of the Maximum Power Release or the value of the Outflow minus the Spill.

The following calculations are not performed if Energy is Input, set by a Rule, or flagged **BEST EFFICIENCY** or **MAX CAPACITY**. In these cases the Power and Energy have already been calculated in Plant Efficiency Curve Release.

If the Power Coefficient is input by the user,

$$Power = TurbineRelease \times PowerCoefficient$$

Otherwise, Power is found directly from the Plant Power Table using the current Operating Head and the Turbine Release from above. The power coefficient is now calculated as:

$$PowerCoefficient = Power / TurbineRelease$$

If the user has input the Plant Power Limit, the following steps are taken:

1. If the Power Coefficient is input by the user, Power and Turbine Release may need to be recalculated. If the Power is greater than the Plant Power Limit, Power is set equal to the Plant Power Limit and Turbine Release is recalculated as the Plant Power Limit divided by the Power Coefficient.

2. If the Power Coefficient is not input and the Plant Power Limit is exceeded; the Turbine Release, Power, and Power Coefficient need to be recalculated. The Power is set equal to the Plant Power Limit and the Turbine Release is found using 3-D interpolation of the Plant Power Table. The Power Coefficient is then calculated as Power divided by Turbine Release.

Energy is then calculated as Power multiplied by the timestep length.

The following calculations take always take place, regardless of the flag on Energy:

If either the Turbine Release is equal to the maximum power release or the Energy is at the maximum capacity, Hydro Capacity is set equal to Power and the Best Hydro Capacity is computed iteratively. If the energy is at the **Best Efficiency**, the Best Hydro Capacity is set equal to the Power and the Hydro Capacity is computed iteratively. If the Turbine Release is not equal to the maximum power release, the Energy is not at the maximum capacity, and the Energy is not at the **Best Efficiency**, both Hydro Capacity and Best Hydro capacity are computed by an iterative algorithm.

Notes on Power Plant Cap Fraction

If the Power Plant Cap Fraction is input by the user, it is necessary for the Plant Power Table to basically be scaled back to account for the operating points when the turbines are operating at less than 100%. To do this, when Turbine Release is known and Power is to be found using the Plant Power Curve, Turbine Release is divided by the Power Plant Cap Fraction. This point is then found in the Plant Power Curve for the current operating head and the Power is found using 3-D interpolation. Finally the Power is multiplied by the Power Plant Cap Fraction to get the actual Power produced for the current timestep.

If Power is known, and Turbine release is to be found in the table. Power is multiplied by the Power Plant Cap Fraction and then this point is found in the Plant Power Curve to solve for Turbine Release. Turbine Release is then divided by the Power Plant Cap Fraction to get the actual Turbine Release for the current timestep.

17.1.1.5 Plant Power Equation

The Plant Power Equation method is used to calculate Power and Energy using the water power equation.

Slots Specific to this Method

☛ Head Loss

Type: Table Slot
Units: LENGTH
Description: The head loss water incurs before it reaches the turbines.
Information: The slot is set to zero if not input by the user.
I/O: optional
Links: Not linkable

☛ Minimum Power Elevation

Type: Table Slot
Units: LENGTH
Description: Minimum pool elevation at which power can be generated
Information: Single value in a 1x1 table slot
I/O: Required input
Links: Not linkable

☛ Net Head vs Max Turbine Release

Type: Table Slot
Units: LENGTH VS. FLOW
Description: relationship between the net head and the maximum possible turbine release
Information: Net Head must account for any head loss
I/O: Required input
Links: Not linkable

☛ Net Head vs Plant Efficiency

Type: Table Slot
Units: LENGTH VS NONE
Description: This table allows you to specify the efficiency as a function of (previous) Net Head.
Information: This table is used only when the Plant Efficiency Value is empty. The Net Head used in this table look up comes from the previous timestep's operating head.
I/O: Optional Input
Links: Not Linkable

☛ Plant Efficiency Value

Type: Table Slot
Units: NONE
Description: the decimal percent efficiency at which the plant is operating
Information: Single value in a 1x1 table slot. Plant efficiency should incorporate both generator efficiency and turbine efficiency.
I/O: Optional Input, if specified, it must be between 0 and 1.
Links: Not linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: NONE
Description: decimal fraction of the power capacity at which the plant is operating
Information: Used in the case of outages or reductions in the plant operating capacity.
I/O: Defaults to 1.0 if not input.
Links: Not linkable

☛ Plant Power Limit

Type: Series Slot
Units: POWER
Description: The user specified upper limit on power production

Information: If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit and the Energy is recalculated. A new Turbine Release are then calculated based on the Plant Power Limit.
I/O: Optional Input or set by a rule.

The method first checks whether Energy is either user input or set by rules. If it is, the method finishes successfully and exits-- all power calculations were already performed in the Plant Power Equation Release method.

Otherwise, Tailwater Elevation and Operating Head are determined based on the user method selected in the Tailwater category.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the Minimum Power Elevation was input by the user. If no value was input, a **RiverWare**™ error is posted and the simulation run is aborted. If the previous Pool Elevation is less than the Minimum Power Elevation or the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Energy and Power are set equal to zero.
2. If the Turbine Release is input or already set from the Dispatch Method "solveMB_givenInflowRelease," a **RiverWare**™ error is flagged and the run is aborted.
3. Turbine Release is set equal to zero.

Once the initial checks are performed, the method calculates Net Head and Turbine Release as:

$$NetHead = OperatingHead - HeadLoss$$

Next, the Max Turbine Release for the current Net Head is interpolated from the Net Head vs. Max Turb Release table.

If there is a valid value in the Plant Efficiency Value slot, that is used for the *efficiency.* Otherwise, given the net head from the previous timestep (Operating Head at previous timestep minus Head Loss), the *efficiency* is interpolated from the Net Head vs Efficiency table. The previous Operating Head is used as an approximation so as not to introduce an additional variable in the iteration. As a result, the Tailwater Elevation at the initial timestep must be input. The net head for the initial timestep is the initial Pool Elevation minus the initial Tailwater Elevation minus Head Loss.

The method checks whether Turbine Release is user input or set by a link or a rule. If Turbine Release is known at the dispatch level, the method will check that it is not greater than Outflow minus Spill or the Max Turbine Release given the current Net Head. If either of these are true, an error will be posted and the run will abort. Otherwise, the known Turbine Release value will be used in the Power calculations. If Turbine Release is **not** user input or solved for in the dispatch methods, it is calculated as the minimum of the Max Turbine Release (given the current Net Head) and Outflow minus Spill (either unregulated spill or user specified regulated spill):

$$TurbineRelease = Min(MaxTurbineRelease \times PowerPlantCapFraction, Outflow - Spill)$$

Note: If Turbine Release is set to MaxTurbineRelease, it means there is still some remaining water that must be passed via regulated spill. This will be calculated in the spill calculations. If Turbine Release is set to Outflow minus Spill, it means that Spill consists of Unregulated Spill and any input Regulated Spill-- all other water will pass through the turbines.

Once efficiency, Net Head, and Turbine Release are all known, Power is solved for using the Power Equation:

$$Power = \frac{\text{Turbine Release} \times \text{Net Head} \times \text{efficiency}}{\text{Unit Compatibility Factor}}$$

The unit compatibility factor comes from balancing units and is 102.01697767 in internal RiverWare units.

If the computed Power is greater than the Plant Power Limit, the Power is reset to the Plan Power Limit and Turbine Release is recomputed by solving the above equation for Turbine Release.

Lastly, Energy is computed as Power multiplied by the time length of the timestep.

$$Energy = \text{Power} \times \text{Length of Timestep}$$

17.1.1.6 Unit Generator Power

The Unit Generator Power method is used to calculate Power and Energy for generating units with individual characteristics. The generating units are grouped by unit type for ease of data entry.

Slots Specific to this Method

☛ **Best Generator Flow**

Type: Table Slot
Units: LENGTH VS. FLOW
Description: a table for each unit type which gives the relationship between operating head and flow through the generator when operating at best efficiency

Information: There must be a block of data for each unit type given in the Generator Unit Types table. The table is representative of a single unit within the specified unit type.
I/O: Required input
Links: Not linkable

BEST GENERATOR POWER
Type: Table Slot
Units: LENGTH VS. POWER
Description: a table for each unit type which gives the relationship between operating head and power produced by the generator when operating at best efficiency
Information: There must be a block of data for each unit type given in the Generator Unit Types table. The table is representative of a single unit within the specified unit type.
I/O: Required input
Links: Not linkable

FULL GENERATOR FLOW
Type: Table Slot
Units: LENGTH VS. FLOW
Description: a table for each unit type which gives the relationship between operating head and flow through the generator when operating at full capacity
Information: There must be a block of data for each unit type given in the Generator Unit Types table. The table is representative of a single unit within the specified unit type.
I/O: Required input
Links: Not linkable

FULL GENERATOR POWER
Type: Table Slot
Units: LENGTH VS. POWER
Description: a table for each unit type which gives the relationship between operating head and power produced by the generator when operating at full capacity
Information: There must be a block of data for each unit type given in the Generator Unit Types table. The table is representative of a single unit within the specified unit type.
I/O: Required input
Links: Not linkable

GENERATOR UNIT TYPES
Type: Table Slot
Units: NONE

Description: a list of each generating unit and the corresponding unit type
Information: More than one generating unit can be assigned to a given unit type. The unit type must be an integer value beginning with 1 and increasing by increments of 1.
I/O: Required input
Links: Not linkable

☛ Generators Available and Limit
Type: Table Series Slot
Units: FRACTION AND POWER
Description: a time series specifying the availability and power limit of each generating unit.
Information: Availability is a number between 0 and 1 which represents the percentage of the timestep that the unit is available. There must be a block of data for each row in the Generator Unit Types table. The Power Limit has no effect on the flow through the turbines.
I/O: Required input
Links: Not linkable

☛ Hydro Capacity
Type: Agg Series Slot
Units: POWER
Description: the maximum power production possible at the current timestep
Information: This value is the sum of all generators operating at full capacity for the given operating head at the current timestep.
I/O: Output only
Links: Could be linked to a Data Object, but usually not linked.

☛ Minimum Power Elevation
Type: Table Slot
Units: LENGTH
Description: the minimum pool elevation required for power production
Information: When the Pool Elevation drops below this value, a warning is posted and no power is produced.
I/O: Required input
Links: Not linkable

☛ Power Coefficient
Type: Series Slot
Units: POWER/FLOW
Description: power generated per unit power release

Information: This coefficient corresponds to the efficiency of the entire plant. It is not used in calculation and is displayed only for the benefit of the user.
I/O: Output only
Links: Could be linked to a Data Object, but usually not linked.

The Unit Generator Power method begins by computing the availability and power limits of each unit type. Availability and power limit values are computed as the sum of the values from the availability and power limit columns, respectively, in the Generators Available and Limit slot. A value for availability and power limit is computed for each unit type.

First, Tailwater Elevation and Operating Head are determined based on the user method selected in the Tailwater category.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction if necessary (the default is 1.0) and checks for plant shutoff/failure.

If the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Energy and Power are set equal to zero.
2. If the Turbine Release is input or already set from the Dispatch Method "solveMB_givenInflowRelease," a **RiverWare**™ error is flagged and the run is aborted.
3. Turbine Release is set equal to zero.

Then the efficiency of each unit type is calculated by the following equation:

$$efficiency = \frac{powerTemp}{flowTemp}$$

PowerTemp and flowTemp, both local variables, are computed from the Best Generator Power and Best Generator Flow tables, respectively, using the current Operating Head. Each unit type is then sorted in descending order based on the computed efficiency.

Turbine Release has already been computed by the dispatch method and the power produced from the known Turbine Release must be calculated. The method begins to add entire unit types (operating according to the best flow and power tables and beginning with the most efficient type) until the Turbine Release is exceed or all the unit types have been added. If the power generated by a particular unit type exceeds the power limit for that unit type, the power produced from that type is set to the power limit. The power limit has no effect on the flow going through the turbines. If the Turbine Release is exceeded, the last generator type is interpolated to compute the Power exactly (see equation below). However, if all the unit types have been added and the Turbine Release cannot be met, the method assumes all unit types are operating at full capacity (according to the Full Generator Flow and Full Generator

Power tables). Then if the Turbine Release is exceeded, the last generator type added is interpolated to compute the Power exactly (see equation below). However, if the Turbine Release still cannot be met, all unit types are run at full capacity. Turbine Release is reset to the maximum flow through the turbines and Power is set as the maximum power produced by the turbines (at the given operating head). The spill must be recalculated to handle the excess Turbine Release that could not be met.

The interpolation equation used to calculate Power is given below:

$$\text{Power} = \text{oneLessTypePower} + \frac{\text{Turbine Release} - \text{oneLessTypeFlow}}{\text{cumulativeFlow} - \text{oneLessTypeFlow}} \bullet (\text{cumulativePower} - \text{oneLessTypePower})$$

where oneLessTypePower is the power produced from all the previous types added (excluding the most recent type added); oneLessTypeFlow is the flow through all the previous unit types (excluding the most recent type added); cumulativePower is the power produced from all the unit types added (including the most recent type); and cumulativeFlow is the flow through all the unit types added (including the most recent type).

Note: The above equation assumes the relationship between power and flow is linear, regardless of the actual relationship specified in the power and flow tables. It is also interpolating over an entire type of generators.

The Power Coefficient is then calculated by the following equation:

$$\text{Power Coefficient} = \frac{\text{Power}}{\text{Turbine Release}}$$

If all the unit types were added, the Spilled Energy Power Coefficient is equal to the Power Coefficient. However, if all the types were not added, the Spilled Energy Power Coefficient is set equal to the efficiency of the last type added.

Energy is calculated as the product of the Power and the timestep length. Hydro Capacity is set as the power produced from all units operating at full capacity.

17.1.1.7 Peak and Base

The Peak and Base method computes the Power and Energy generated by the entire plant based on the fraction of each timestep operated at peak flow and base flow. It is a long timestep method, modeled after the U. S. Bureau of Reclamation’s CRSS peak-base power calculation. A peaking flow value is first determined from the Outflow, Tailwater Elevation and Best Generator Flow. A minimum Base Flow and Power production are assumed for the entire timestep. Next, the number of hours to operate at peak power is calculated from the remaining volume of water released during that timestep. Peak production and base production are then added to determine the total Energy. Power is calculated by dividing the

Energy by the timestep length in hours. Power Capacity is the power that could be generated if the flow is directed through the turbine(s) given an operating head. This is added to distinguish between actual power production and the power that could be produced.

Slots Specific to this Method

☛ Base Flow

Type: Series Slot
Units: FLOW
Description: minimum flow through turbines to produce energy
Information: This value is read from the Base Flow Table
I/O: Output only
Links: Not linkable

☛ Base Flow Table

Type: Table Slot
Units: FLOW vs. FLOW
Description: Outflow from the Reservoir vs. base flow
Information: This table gives the minimum flow required through the turbines as a function of the average total outflow from the Reservoir.
I/O: Required input
Links: Not linkable

☛ Best Generator Flow

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Operating Head vs. flow through the turbine at best efficiency
Information: The minimum and maximum values of operating head in this table are used as limiting values. The operating head is reset to the min. or max if it exceeds these constraints.
I/O: Required input
Links: Not linkable

☛ Best Generator Power

Type: Table Slot
Units: LENGTH vs. POWER
Description: Operating Head vs. power at best efficiency
Information: power produced by the entire plant at base energy flow
I/O: Required input
Links: Not linkable

☛ Maximum Turbine Power

Type: Table Slot
Units: POWER
Description: maximum turbine power output
Information:
I/O: Required input
Links: Not linkable

☛ Min and Max Operating Head

Type: Table Slot
Units: LENGTH
Description: the minimum and maximum operating head for the turbines
Information:
I/O: Required input
Links: Not linkable

☛ Number of Units

Type: Table Slot
Units: NONE
Description: integer number of turbines in plant
Information:
I/O: Required input
Links: Not linkable

☛ Off Peak Capacity

Type: Table Slot
Units: POWER
Description:
Information:
I/O: Required input
Links: Not linkable

☛ Peak Flow

Type: Series Slot
Units: FLOW
Description: most efficient flow through turbines for the current Operating Head
Information:
I/O: Output only
Links: Not linkable

☛ Peak Hours

Type: Series Slot
Units: TIME
Description: the number of hours operated at peak flow
Information:
I/O: Output only
Links: Not linkable

☛ Plant Efficiency

Type: Series Slot
Units: FRACTION
Description: ratio of actual power produced to peak and base theoretical power
Information:
I/O: Output only
Links: Not linkable

☛ Power Capacity

Type: Agg Series Slot
Units: POWER
Description: power that could be produced if flow is directed through the turbines given the operating head.
Information: Calculated by the two peak power methods and cannot be input by the user.
I/O: Output only
Links: Not linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: FRACTION
Description: the percentage of full capacity of the turbine units in the hydropower plant
Information:
I/O: Optional; the value of this slot defaults to 100% if not input by user
Links: Not linkable

First, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction if necessary (the default is 1.0) and checks for plant shutoff/failure.

If the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Energy, Power, Base Flow, Peak Flow, Peak Power, Power Capacity, and Plant Efficiency are set equal to zero.

2. If the Turbine Release is input or already set from the Dispatch Method "solveMB_givenInflowRelease," a **RiverWare**™ error is flagged and the run is aborted.

3. Turbine Release is set equal to zero.

4. Tailwater and Operating Head are computed and set. No further computations are performed.

If either the Energy or Turbine Release is input by the user, a **RiverWare**™ error is posted and the simulation run is aborted. These are not valid input slots for the Peak and Base method. Peak Flow and Peak Power are then calculated as follows:

1. If either the Maximum Turbine Power is not valid or it is less than 0.00000001 MW, a **RiverWare**™ Error is flagged and the simulation run is aborted.

2. The Maximum Turbine Power is then used with the Best Generator Power table to obtain a value for the local variable, headAtMaxPower.

3. headAtMaxPower is used with the Best Generator Flow table to obtain a value for the local variable, flowAtMaxPower. The local variable, flow is temporarily set as flowAtMaxPower.

4. The local variable, efficiencyAtMaxPower is calculated by the following formula:

$$\text{efficiencyAtMaxPower} = \text{Maximum Turbine Power} \times \frac{1000000}{\text{flowAtMaxPower} \times \text{headAtMaxPower} \times 999.99 \times 9.79908}$$

where 999.99 is the density of water (Kg/M^3) at five degrees C and 9.79908 is gravitational acceleration (M/s^2) at 37 degrees North latitude.

5. **RiverWare**™ then iterates while:

- The absolute difference between Qnew and flow is greater than 5 cfs.
- The number of iterations is less than the maximum number of iterations.
- The Operating Head is greater than the Minimum Operating Head.

The following calculations and evaluations are inside the iterative loop:

- The local variable, Qnew is set equal to flow
- The local variable, plantFlow is determined using the following equation:

$$\text{plantFlow} = \text{Qnew} \times \text{Number of Units} \times \text{Power Plant Cap Fraction}$$

- Flow is set equal to plantFlow
- The user selected Tailwater calculation is performed
- The Operating Head is calculated
- If the Operating Head is greater than the Maximum Operating Head, Operating Head is set equal to the maximum Operating Head
- If the Operating Head is less than the Minimum Operating Head, Operating Head is set equal to the Minimum Operating Head
- If the Operating Head is greater than the headAtMaxPower, the flow is calculated using the following equation:

$$\text{flow} = \text{Maximum Turbine Power} \times \frac{1000000}{\text{Operating Head} \times \text{efficiencyAtMaxPower} \times 999.99 \times 9.79908}$$

- If the Operating Head is less than the Maximum Operating Head, greater than the Minimum Operating Head, and less than the headAtMaxPower, flow is obtained from the Best Generator Flow table and the Operating Head.

This set of calculations is repeated until the iteration criteria described above are met.

6. If Operating head is less than the Minimum Operating Head; Turbine Release, Energy, Power, Power Capacity, Peak Flow, Peak Hours, and flow are all set to zero. Then the Tailwater method is re-executed. If Operating Head is greater than the minimum Operating Head and the headAtMaxPower, Peak Power is set equal to Maximum Turbine Power. If the Operating Head is greater than the Minimum Operating Head and less than the headAtMaxPower, Peak Power is determined from the Best Generator Power table using Operating Head.

Once Peak Power and Peak Flow (called “flow” in calculations described above) are calculated, Base Power and Base Flow can be determined. Base Power is set equal to Off Peak Capacity and Base Flow is determined from Outflow and the Base Flow Table.

If Outflow minus Unregulated Spill is greater than the product of Peak Flow, Number of Units, and Power Plant Cap Fraction; the following steps are taken:

1. The Tailwater method selected by the user is executed.
2. The Operating Head is calculated.
3. The local variable, headAtMaxPower, is obtained from the Best Generator Power table using the Maximum Turbine Power.

4. The local variable, flowAtMaxPower, is obtained from the Best Generator Flow table using the headAtMaxPower.

5. The local variable, efficiencyAtMaxPower, is computed using the following formula:

$$\text{efficiencyAtMaxPower} = \frac{\text{Maximum Turbine Power} \times 1000000}{(\text{flowAtMaxPower} \times \text{headAtMaxPower} \times 999.99 \times 9.79908)}$$

6. If Operating Head is greater than headAtMaxPower, Peak Flow and Peak Power are calculated using the following equations:

$$\text{Peak Flow} = \text{Maximum Turbine Power} \times \frac{1000000}{\text{Operating Head} \times \text{EfficiencyAtMaxPower} \times 999.99 \times 9.79908}$$

$$\text{Peak Power} = \text{Maximum Turbine Power}$$

7. If the Operating Head is less than or equal to the headAtMaxPower, Peak Flow and Peak Power are determined using the Operating Head in conjunction with the Best Generator Flow and Best Generator Power tables, respectively.

The Peak Flow slot represents the flow through the entire power plant. Therefore, the value in this slot is calculated as Peak Flow times Number of Units times Power Plant Cap Fraction. In the calculations that follow, Peak Flow represents the slot value just calculated. Plant Peak Power is calculated as Peak Power times Number of Units times Power Plant Cap Fraction.

The number of hours required to operate at base and peak flows are computed next using the following equations:

$$\text{Peak Flow Volume} = (\text{Outflow} - \text{Spill} - \text{Base Flow}) \times \text{timestep (in seconds)}$$

$$\text{Peak Hours} = \frac{\text{Peak Flow Volume}}{(\text{Peak Flow} - \text{Base Flow}) \times 3600}$$

$$\text{Base Hours} = \frac{\text{Timestep (in seconds)}}{3600} - \text{Peak Hours}$$

If Peak Hours is greater than the length of the timestep; Peak Hours is set equal to the timestep, Base Hours are set to zero, Turbine Release is set to Peak Flow, and Total Controlled Release is calculated as Outflow minus Unregulated Spill. If Peak Hours is less than or equal to the length of the timestep, Peak Hours and Base Hours remain as calculated by the above formulas, Turbine Release is Outflow minus Spill, and Total Controlled Release is set equal to the Peak Flow.

The theoretical and actual energy production are computed next. The local variable, peakEnergy is calculated as Peak Hours times Plant Peak Power. The local variable, baseEnergy is calculated as Base Hours times Base Power.

The local variable, bestBaseTheor (representing the theoretical, most efficient base energy) is calculated using the following equation:

$$\text{bestBaseTheor} = \text{Base Flow} \times \text{Operating Head at Base Flow} \times \text{Base Hours} \times 0.00980229$$

The local variable bestPeakTheor (representing the theoretical most efficient peak energy) is calculated using the following equation:

$$\text{bestPeakTheor} = \text{Peak Flow} \times \text{Operating Head at Peak Flow} \times \text{Peak Hours} \times 0.00980229$$

The value, 0.00980229, is a conversion factor necessary for energy to have units of megawatt-hours.

$$\text{Energy (MWH)} = 62.4\ (\text{lb/ft3}) \times Q\ (\text{cfs}) \times \text{time (hrs)} \times \text{Head (ft)} \times 1\ \text{KW} / 737.56\ (\text{ft-lb/s}) \times 1\ \text{MW} / 1000\ \text{KW}$$

For Q in cms and Head in meters the final conversion is 0.00980229.

Finally, the following slots are set:

$$\text{Power Capacity} = \text{Plant Peak Power}$$

$$\text{Energy} = \text{peakEnergy} + \text{baseEnergy}$$

$$\text{Power} = \frac{\text{Energy}}{\text{timestep length (hours)}}$$

$$\text{Plant Efficiency} = \frac{\text{Energy}}{\text{bestPeakTheor} + \text{bestBaseTheor}}$$

If a spill method is selected which utilizes the Spilled Energy Power Coefficient and this value is not input by the user, it is set as:

$$\text{Spilled Energy Power Coefficient} = \frac{\text{Power}}{\text{Turbine Release}}$$

If Turbine Release is zero, the Spilled Energy Power Coefficient is also zero.

17.1.1.8 Peak Power

The Peak Power method is similar to the Peak and Base method except that it computes power and energy based on Peak Flow only. A peaking flow value is first determined from the Outflow, Tailwater Elevation, and Best Generator Flow. The number of hours to operate at peak power is then calculated from the volume of water released during that timestep. A distinction is made between actual power production and the power that could be produced. Power Capacity is the peak power capacity. Power is calculated by dividing the energy by the timestep length in hours. There is no Base Flow power production.

Slots Specific to this Method

☛ **Best Generator Flow**

Type: Table Slot
Units: LENGTH VS. FLOW
Description: operating head vs. flow through the turbine at best efficiency
Information: The minimum and maximum values of operating head in this table are used as limiting values. The operating head is reset to the min. or max if it exceeds these constraints.
I/O: Required input
Links: Not linkable

☛ **Best Generator Power**

Type: Table Slot
Units: LENGTH VS. POWER
Description: operating head vs. power at best efficiency
Information:
I/O: Required input
Links: Not linkable

☛ **Generator Efficiency**

Type: Table Slot
Units: FRACTION
Description: the efficiency of the generators in producing power
Information: This value is the fraction of the maximum theoretical power which could be obtained from an ideal turbine.
I/O: Required input
Links: Not linkable

☛ **Maximum Turbine Power**

Type: Table Slot
Units: POWER

Description: maximum turbine power output
Information:
I/O: Required input
Links: Not linkable

☛ Min and Max Operating Head

Type: Table Slot
Units: LENGTH
Description: the minimum and maximum operating head for the turbines
Information:
I/O: Required input
Links: Not linkable

☛ Number of Units

Type: Table Slot
Units: NONE
Description: integer number of turbines in plant
Information:
I/O: Required input
Links: Not linkable

☛ Peak Flow

Type: Series Slot
Units: FLOW
Description: most efficient flow through turbines for the current Operating Head
Information:
I/O: Output only
Links: Not linkable

☛ Peak Hours

Type: Series Slot
Units: TIME
Description: the number of hours operated at peak flow
Information:
I/O: Output only
Links: Not linkable

☛ Plant Efficiency

Type: Series Slot
Units: FRACTION
Description: ratio of actual power produced to peak and base theoretical power

Information:
I/O: Output only
Links: Not linkable

☛ Power Capacity

Type: Agg Series Slot
Units: POWER
Description: power that could be produced if flow is directed through the turbines given the operating head.
Information: Calculated by the two peak power methods and cannot be input by the user.
I/O: Output only
Links: Not linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: FRACTION
Description: the percentage of full capacity of the turbine units in the hydropower plant
Information: The value of this slot defaults to 100% if not input by user.
I/O: Required input
Links: Not linkable

First, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction if necessary (the default is 1.0) and checks for plant shutoff/failure.

If the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Energy, Power, Peak Flow, Peak Hours, Power Capacity, and Plant Efficiency are set equal to zero.
2. Turbine Release is set equal to zero.
3. Tailwater and Operating Head are computed and set. No further computations are performed.

If Energy or Turbine Release are input by the user, an error is posted. These are not valid input slots for the Peak Power method. Peak Flow and Peak Power are then calculated as follows:

1. The Maximum Turbine Power is then used in the Best Generator Power table interpolate a value for headAtMaxPower.
2. headAtMaxPower is used in the Best Generator Flow table to interpolate a value for flowAtMaxPower. Flow is temporarily set as flowAtMaxPower.

3. efficiencyAtMaxPower is calculated by the following formula:

$$\text{efficiencyAtMaxPower} = \text{Maximum Turbine Power} \times \frac{1000000}{\text{flowAtMaxPower} \times \text{headAtMaxPower} \times 999.99 \times 9.79908}$$

where 999.99 is the density of water (Kg/M^3) at five degrees C and 9.79908 is gravitational acceleration (M/s^2) at 37 degrees North latitude.

4. **RiverWare**™ then iterates until all of the following conditions are met; the absolute difference between Qnew and flow is less than 5 cfs, the number of iterations is greater than the maximum number of iterations, and the Operating Head is less than the Minimum Operating Head. Initially, Qnew is set equal to flow. Plant Flow is calculated as Qnew times the Number of Units times the Plant Power Cap Fraction. Then, the Tailwater method selected by the user is performed. If the Operating Head is greater than the Maximum Operating Head, the Operating Head is set equal to the Maximum Operating Head. If the Operating Head is less than the Minimum Operating Head, Operating Head is set equal to Minimum Operating Head. If the Operating Head is greater than headAtMaxPower, flow is calculated as:

$$\text{flow} = \text{Maximum Turbine Power} \times \frac{1000000}{\text{Operating Head} \times \text{efficiencyAtMaxPower} \times 999.99 \times 9.79908}$$

Otherwise, flow is determined by interpolation using Operating Head and the Best Generator Flow table. This set of calculations is repeated until the iteration criteria described above are met.

5. If Operating head is less than the Minimum Operating Head, Turbine Release, Energy, Power, Power Capacity, Peak Flow, Peak Hours, and flow are all set to zero. Then, the Tailwater method is re-executed. If Operating Head is greater than headAtMaxPower, Peak Power is set equal to Maximum Turbine Power. Otherwise, Peak Power is determined from the Best Generator Power table using Operating Head.

Once flow and Peak Power are determined, the following computations are performed:

$$\text{Peak Flow} = \text{flow} \times \text{number of Units} \times \text{Power Plant Cap Fraction}$$

$$\text{Plant Peak Power} = \text{Peak Power} \times \text{Number of Units} \times \text{Power Plant Cap Fraction}$$

$$\text{Peak Flow Volume} = (\text{Outflow} - \text{Spill}) \times \text{Timestep (in seconds)}$$

$$\text{Peak Hours} = \frac{\text{Peak Flow Volume}}{\text{Peak Flow} \times 3600 \text{ seconds}}$$

If the value of Peak Hours is greater than the length of the timestep, Peak Hours is set to the length of the timestep, Turbine Release is equal to Peak Flow, and Total Controlled Release is equal to Outflow minus Unregulated Spill. Otherwise, Peak Hours remains unchanged, Turbine Release equals Outflow minus spill, and Total Controlled Release equals Peak Flow.

Next, the theoretical and actual energy production is calculated.

$$\text{Peak Energy} = \text{Peak Hours} \times \text{Plant Peak Power} \times \text{Generator Efficiengy}$$

$$\text{Best Peak Theoretical} = \text{Peak Flow} \times \text{Operating Head} \times \text{Peak Hours} \times 0.00980229$$

The value, 0.00980229, is a conversion factor necessary for energy to have units of megawatt-hours.

$$\text{Energy (MWH)} = 62.4\ (\text{lb/ft3}) \times Q\ (\text{cfs}) \times \text{time (hrs)} \times \text{Head (ft)} \times 1\ \text{KW} / 737.56\ (\text{ft-lb/s}) \times 1\ \text{MW} / 1000\ \text{KW}$$

For Q in cms and Head in meters the final conversion is 0.00980229.

Finally, the following slots are set:

$$\text{Power Capacity} = \text{Plant Peak Power}$$

$$\text{Energy} = \text{Peak Energy}$$

$$\text{Power} = \frac{\text{Energy}}{\text{timestep length (hours)}}$$

If Best Peak Theoretical is equal to zero, Plant Efficiency is also equal to zero. Otherwise Plant Efficiency is calculated as:

$$\text{Plant Efficiency} = \frac{\text{Energy}}{\text{Best Peak Theoretical}}$$

If a spill method is selected which utilizes the Spilled Energy Power Coefficient and this value is not input by the user, it is set as:

$$\text{Spilled Energy Power Coefficient} = \frac{\text{Power}}{\text{Turbine Release}}$$

If Turbine Release is zero, the Spilled Energy Power Coefficient is also zero.

17.1.1.9 Peak Power Equation

The Peak Power Equation method provides a standard equation method of calculating plant peaking power for a portion of the computational timestep using the water power equation.

Slots Specific to this Method

☛ Head Loss

Type: Table Slot
Units: LENGTH
Description: The head loss water incurs before it reaches the turbines.
Information: The slot is set to zero if not input by the user.
I/O: optional
Links: Not linkable

☛ Min Power Elevation

Type: Table Slot
Units: LENGTH
Description: Minimum pool elevation at which power can be generated
Information: Single value in a 1x1 table slot
I/O: Required input
Links: Not linkable

☛ Net Head vs. Peak Release

Type: Table Slot
Units: LENGTH VS. FLOW
Description: relationship between the net head and the maximum possible turbine release
Information: Net Head must account for any head loss
I/O: Required input
Links: Not linkable

☛ Peak Release

Type: Series Slot
Units: FLOW
Description: The flow through the turbines when the plant is operating at peak capacity
Information: Peak Release is solved for iteratively using net head, tailwater elevation and pool elevation
I/O: Output only
Links: Not linkable

☛ Peak Times

Type: Series Slot
Units: TIME
Description: The time at which the plant is operating at peak capacity
Information: Peak Time is calculated as the timestep flow volume divided by the Peak Release.

I/O: Output
Links: Not linkable

☛ Plant Efficiency

Type: Table Slot
Units: NONE
Description: the decimal percent efficiency at which the plant is operating
Information: Single value in a 1x1 table slot. Plant efficiency should incorporate both generator efficiency and turbine efficiency.
I/O: Defaults to 1.0 if not input. Must be between 0 and 1.
Links: Not linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: NONE
Description: decimal fraction of the power capacity at which the plant is operating
Information: Used in the case of outages or reductions in the plant operating capacity.
I/O: Defaults to 1.0 if not user input. Must be between 0 and 1.
Links: Not linkable

The Peak Power Equation method first performs a series of checks.

The selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the Minimum Power Elevation was input by the user. If no value was input, a **RiverWare**™ error is posted and the simulation run is aborted. If the previous Pool Elevation is less than the Minimum Power Elevation or the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Power, Energy, Peak Release, and Peak Time are set equal to zero.
2. If the Turbine Release is input or already set from the Dispatch Method “solveMB_givenInflowRelease,” a **RiverWare**™ error is flagged and the run is aborted.
3. Turbine Release is set equal to zero.
4. Tailwater and operating head are computed. No further calculations are performed.

The method checks whether Turbine Release (the average flow through the turbines over the whole timestep) is user input or set by rules. (i.e. the method checks if the dispatch type is solveMB_givenInflowRelease.) If the given Turbine Release value is greater than Outflow minus Spill, an error is posted and the run is aborted.

If Turbine Release is not user input or set by rules, it is calculated as the minimum of MaxTurbine Release and Outflow minus Spill,

$$TurbineRelease = Min(MaxTurbineRelease, Outflow - Spill)$$

where the Max Turbine Release is interpolated from the Peak Release vs. Net Head Table, given the Net Head over the entire timestep.

In order to calculate the time at peak production, the flow which passes through the turbines during this time period must be calculated. This Peak Release is the maximum possible flow through the turbines given the Net Head and will be solved for iteratively as described in the steps below.

1. Peak Release is initially set to zero.
2. Tailwater Elevation is determined using Peak Release + Spill as the “flow” value in the selected Tailwater method. (If the Turbine Release slot is linked, it can be assumed that Spill is sent elsewhere and does not affect Tailwater so the “flow” value should be set to Peak Release only.)
3. The operating head is calculated as the Pool Elevation minus the Tailwater Elevation.
4. The Net Head is calculated as the operating head minus the head loss.
5. Given the Net Head, the Peak Release is interpolated from the Net Head vs. Peak Release table and then multiplied by the Power Plant Cap Fraction.
6. The new Peak Release value is compared with the previous value and the iteration will continue until the value converges. (Note: Convergence Percentage is a general slot on power reservoirs representing the convergence in all iterative solutions-- the slot defaults to 0.0001 if not input.)

Once Peak Release is calculated, the Peak Time will be solved for as the volume of flow that passes through the turbines in a timestep divided by the Peak Release:

$$\text{Peak Time} = \frac{\text{Timestep Flow Volume}}{\text{Peak Relase}}$$

where timestepFlowVolume is an internal variable defined as:

$$\text{Timestep Flow Volume} = \text{Turbine Release} \times \text{timestep seconds}$$

RiverWare checks that the Peak Time is not greater than the timestep length. If it is, the run aborts and an error is posted.

Power is calculated with the standard water power equation. The Peak Power Equation method uses Peak Release as the flow value and Net Head at Peak Release as the head value.

$$Power = \frac{\text{Peak Release} \times \text{Net Head} \times \text{efficiency} \times \text{Plant Cap Fraction}}{\text{Unit Compatibility Factor}}$$

The unit compatibility factor comes from balancing units and is 102.01697767 in internal RiverWare units.

Energy is finally computed as the product of Power and Peak Time:

$$Energy = Power \times \text{Peak Time}$$

17.1.1.10 Peak Power Equation with Off Peak Spill

The Peak Power Equation with Off Peak Spill method provides a standard equation to calculate peaking power for a portion of the computational timestep using the water power equation. Included also is a calculation of the off peak spill that occurs when the turbines are not operating.

Slots Specific to this Method

☛ Head Loss

Type: Table Slot
Units: LENGTH
Description: The head loss water incurs before it reaches the turbines.
Information: The slot is set to zero if not input by the user.
I/O: optional
Links: Not linkable

☛ Maximum Power Pool Drawdown

Type: Scalar
Units: LENGTH
Description: maximum vertical drop permitted in the power pool in one timestep for power release
Information:
I/O: Required input
Links: Not Linkable

☛ Minimum Power Elevation

Type: Table Slot
Units: LENGTH
Description: Minimum pool elevation at which power can be generated.

Information: Single value in a 1x1 table slot
I/O: Required input
Links: Not linkable

☛ Minimum Elevation for Power Operations

Type: Series Slot with Periodic Input, HERE (Slots.pdf, Section 4.8)
Units: LENGTH
Description: Minimum pool elevation at which power operations can occur.
Information: This slot provides another way to limit the additional proposed hydropower release described HERE (USACE_SWD.pdf, Section 3.9). In simulation, a warning is issued if the Pool Elevation is below this elevation.
I/O: Optional input as either series or periodic values.
Links: Not linkable

☛ Net Head vs Plant Efficiency

Type: Table Slot
Units: LENGTH VS NONE
Description: relationship between the Net Head and the efficiency of the plant
Information: Net Head includes head loss and Efficiency includes both generator and turbine efficiency
I/O: Required Input
Links: Not Linkable

☛ Net Head vs. Generator Capacity

Type: Table Slot
Units: LENGTH VS. POWER
Description: relationship between Net Head and the maximum possible power produced
Information: Net Head includes head loss
I/O: Required Input
Links: Not Linkable

☛ Off Peak Spill

Type: Series Slot
Units: FLOW
Description: The spill that occurs during the off peak portion of the timestep.
Information: Off Peak Spill is the fraction of the spill that occurs when power is not being produced. The time weighted average of Off Peak Spill and Peak Spill equals the Spill. If the Peak Time equals the timestep length, Off Peak Spill is NaN.
I/O: Output only
Links: Not linkable

☛ Peak Release

Type: Series Slot
Units: FLOW
Description: The flow through the turbines when the plant is operating at generator capacity
Information: Peak Release is solved for iteratively using net head, tailwater elevation and pool elevation
I/O: Output only
Links: Not linkable

☛ Peak Time

Type: Series Slot
Units: TIME
Description: The time at which the plant is operating at peak capacity
Information: Peak Time is calculated as the timestep flow volume divided by the Peak Release.
I/O: Output
Links: Not linkable

☛ Plant Power Limit

Type: Series Slot
Units: POWER
Description: The user specified upper limit on power production.
Information: If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit and the Energy is recalculated. A new Turbine Release is then calculated based on the Plant Power Limit.
I/O: Optional Input
Links: Not Linkable

☛ Peak Spill

Type: Series Slot
Units: FLOW
Description: The spill that occurs during the Peak Time.
Information: Peak Spill is the portion of the Spill that occurs during the Peak Time. The time weighted average of Peak Spill and Off Peak Spill equals the Spill.
I/O: Output
Links: Not linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: NONE
Description: decimal fraction of the power capacity at which the plant is operating

Information: Used in the case of outages or reductions in the plant operating capacity.
I/O: Defaults to 1.0 if not user input. Must be between 0 and 1.
Links: Not linkable

This method is called from the dispatch method, typically after Outflow, Storage, and Pool Elevation have been calculated. If Energy is input, then this method is also called from an iterative loop used to determine the Turbine Release, Peak Release, and Peak Time and/or Spill that satisfies the Energy.

The Peak Power Equation with Off Peak Spill method performs a series of checks. First, the method checks if maximum drawdown is exceeded for two cases. In the first case, if there is a valid Top of Conservation Pool slot, i.e. this is a U.S. Army Corp of Engineers model, the method checks if the Pool Elevation is greater than the Top of Conservation Pool. If so, no error or warning is posted; the drawdown limitation only applies to the conservation pool. If the drawdown is exceeded and the Pool Elevation is less than the top of conservation pool, an error is posted and the run is aborted. Note, an abortive error is only posted if the reservoir is dispatching at the current controller timestep. If it is dispatching at a forecast timestep and max power pool drawdown is exceeded, no error is issued. It is assumed that either the inflow or outflow will be modified when the reservoir dispatches at the current timestep and would catch any errors then.

In the second case, if there is no Top of Conservation Pool slot (i.e. a non U.S. Army Corp of Engineers model), if the calculated Pool Elevation results in exceeding the Maximum Power Pool Drawdown, a warning is posted.

If the calculated Pool Elevation is less than the value in the Minimum Elevation for Power Operations, a warning message is posted.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the Minimum Power Elevation was input by the user. If no value was input, a **RiverWare**™ error is posted and the simulation run is aborted. If the previous Pool Elevation is less than the Minimum Power Elevation or the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Power, Energy, Peak Release, and Peak Time are set equal to zero.
2. If the Turbine Release is input or already set from the Dispatch Method "solveMB_givenInflowRelease," a **RiverWare**™ error is flagged and the run is aborted.
3. Turbine Release is set equal to zero.
4. Tailwater and operating head are computed. No further calculations are performed.

The method checks whether Turbine Release (the average flow through the turbines over the whole timestep) is user input or set by rules. (i.e. the method checks if the dispatch type is solveMB_givenInflowRelease.) If so, and if the given Turbine Release value is greater than Outflow minus Spill, an error is posted and the run is aborted.

In order to calculate the time at peak production, the flow which passes through the turbines during this time period must be calculated. This Peak Release is the maximum possible flow through the turbines given the Net Head and will be solved for iteratively as described in the steps below.

1. Peak Release is initially set to zero.
2. Given the net head from the previous timestep (Operating Head at previous timestep minus Head Loss), the efficiency is interpolated from the Net Head vs Efficiency table. The previous Operating Head is used as an approximation so as not to introduce an additional variable in the iteration. As a result, the Tailwater Elevation at the initial timestep must be input. The net head for the initial timestep is the initial Pool Elevation minus the initial Tailwater Elevation minus Head Loss.
3. The current Tailwater Elevation is determined using the maximum of Peak Release plus Unregulated Spill or the current Outflow as the flow value in the selected Tailwater method. (Note that if Energy is input, the Tailwater Elevation slot value shown will be calculated using the average Outflow from the timestep, but the tailwater elevation used in the Peak Release calculation is calculated as described here. If Energy is not input, the Tailwater Elevation slot value will be the peak Tailwater Elevation calculated here.)
4. The Operating Head is calculated as the average Pool Elevation minus the Tailwater Elevation.
5. The Net Head is calculated as the Operating Head minus the Head Loss.
6. Given the Net Head, the Generator Capacity is interpolated from the Net Head vs. Generator Capacity table and is then multiplied by the Power Plant Cap Fraction. If this new capacity is greater than the Plant Power Limit, if valid, the generator capacity is reset to the Plant Power Limit.
7. Peak Release is calculated according to the power equation. The unit compatibility factor comes from balancing units and the specific weight of water; it is 102.01697767 in internal RiverWare units.

$$\text{Peak Release} = \frac{\text{Generator Capacity} \times \text{Unit Compatibility Factor}}{\text{Net Head} \times \text{Efficiency}}$$

The new Peak Release value is compared with the previous value and the iteration, steps 3-7, continue until the value converges. (Note: Convergence Percentage is a general slot on power reservoirs representing the convergence in all iterative solutions-- the slot defaults to 0.0001 if not input.)

If Turbine Release is not user input or set by rules (not in the dispatch method solveMB_givenInflowRelease), TempTurbineRelease is calculated as the minimum of the Peak Release and Outflow minus Spill,

$$TempTurbineRelease = Min(PeakRelease, Outflow - Spill)$$

The Spill will be non-zero only if there is Unregulated Spill or a spill value is set by user input or rules. Once Peak Release is calculated, the Peak Time will be solved for as the volume of flow that passes through the turbines in a timestep divided by the Peak Release:

$$\text{Peak Time} = \frac{\text{TempTurbine Release} \times \text{timestep seconds}}{\text{Peak Release}}$$

RiverWare checks that the Peak Time is not greater than the timestep length. If it is, the run aborts and an error is posted. Next power is set to be the Generator Capacity:

$$Power = GeneratorCapacity$$

Turbine Release is the Peak Release averaged over the whole timestep:

$$TurbineRelease = \frac{PeakRelease \times PeakTime}{TimestepSeconds}$$

Energy is computed as the product of Power and Peak Time:

$$Energy = Power \times \text{Peak Time}$$

Peak Spill and Off Peak Spill are then determined based on Peak Time, Spill, and Unregulated Spill. If Unregulated Spill is non-zero, then Peak Spill is assumed to be equal to Unregulated Spill. Unregulated spill is calculated based on the pool elevation and occurs over the entire timestep. Off Peak Spill is the sum of Unregulated Spill and the Regulated Spill plus Bypass apportioned over the off peak time. If there is no Unregulated Spill, the Peak Spill is zero, and the Off Peak Spill is the Regulated Spill plus Bypass apportioned over the off peak time. If the Peak Time is equal to the timestep length, then Peak Spill is equal to Spill and Off Peak Spill remains NaN.

Finally, if the Load slot is visible and valid, the Thermal Purchase, Dump Energy and Operation Factor are calculated. See the Load Calculation Section on page 624 for more information on these slots.

17.1.1.11 LCR Power

The LCR Power method uses an empirical relationship to calculate the energy produced by the Hoover, Davis, and Parker dams on the Lower Colorado River. The method replicates the calculations from the U.S. Bureau of Reclamation BHOPS FORTRAN program. Energy is calculated as a function of flow, Operating Head, Plant Efficiency, and the Power Coefficients.

Slots Specific to this Method

☛ LCR Input Efficiency

Type: Series Slot
Units: NONE
Description: a fractional value ranging from 0 to 1 which may be used to scale the efficiency or the turbine units in the hydropower plant.
Information:
I/O: Optional; the value defaults to 1 if not input by the user.
Links: Not linkable

☛ Lower Colo Power Coeffs

Type: Table Slot
Units: NONE
Description: two values used as empirical coefficients in relating flow, head, and efficiency to energy.
Information: The coefficients for Hoover Dam are empirically derived. For Davis and Parker dams, they reduce to coeff1=1 and coeff2=0.
I/O: Required input
Links: Not linkable

☛ Net Energy Request

Type: Series Slot
Units: ENERGY
Description: represents the total energy requested by the grid
Information: This slot is only used when a Best Efficiency flag is set in the Energy slot.
I/O: Optional; only used when a Best Efficiency flag is set in the Energy slot.
Links: Not linkable

☛ Plant Efficiency

Type: Series Slot
Units: NONE
Description: a fractional value ranging from 0 to 1 that represents the percentage of full efficiency of the turbine units in the hydropower plant

Information: In the case of Davis and Parker dams, this equals the LCR Input Efficiency. For Hoover Dam, a Plant Efficiency is calculated.
I/O: Output only
Links: Not linkable

☛ Power Coefficient

Type: Series Slot
Units: POWER PER FLOW
Description: power generated per unit power release
Information: This coefficient corresponds to the efficiency of the entire plant. It is not used in calculation and is displayed only for the benefit of the user.
I/O: Output only
Links: Not linkable

☛ Station Energy Table

Type: Table Slot
Units: ENERGY
Description: represents the energy required to run the station for each day of the week
Information: This slot is only used when a Best Efficiency flag is set in the Energy slot.
I/O: Optional; only used when a Best Efficiency flag is set in the Energy slot.
Links: Not linkable

The first step in the LCR Power method is to determine the Operating Head. This is accomplished by executing the Tailwater method specified by the user.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Power, Energy, Plant Efficiency, and Power Coefficient are set equal to zero.
2. If the Turbine Release is input or already set from the Dispatch Method "solveMB_givenInflowRelease," a **RiverWare**™ error is flagged and the run is aborted.
3. Turbine Release is set equal to zero. No further calculations are performed.

If Energy is input by the user or it has been flagged as Best Efficiency, the LCR Power Release method is called to calculate the Turbine Release. This method takes a power request, determines if it can be met given the maximum power that can be generated for the given head, and sets the Turbine Release required to generate the requested power. If the

Turbine Release is not already calculated by the LCR Power Release method, it is set as Outflow minus Spill. The value of Turbine Release is checked against the maximum value set by the user (if it has been set). If Turbine Release is greater than the maximum value an error is posted which reads, “Turbine Release required to meet Energy request is greater than the maximum Turbine Release.”

Power and Energy are then calculated by the following equations:

$$\text{Energy (MWH)} = \left(\text{Lower Colo Power Coeffs\_1} \times \frac{62.4}{737.5} \times \text{Outflow (1000 cfs)} \times \text{Timestep (hours)} \times \frac{\text{Operating Head (ft)}}{1000} - \text{Lower Colo Power Coeffs\_2}\right) \times \text{LCR Input Efficiency} \times 1000$$

where 62.4 is the unit weight of water in pounds per cubic foot and 737.5 represents ft.-lb./sec. per Kilowatt.

$$\text{Power} = \frac{\text{Energy}}{\text{Timestep (hours)}}$$

If energy is zero, Plant Efficiency and the Power Coefficient are also zero. Otherwise they are calculated as:

$$\text{Plant Efficiency} = \frac{\text{Energy}/1000}{\frac{62.4}{737.5} \times \text{Outflow (1000 cfs)} \times \text{Timestep (hours)} \times \frac{\text{Operating Head (ft)}}{1000}}$$

$$\text{Power Coefficient} = \frac{\text{Energy}}{\text{Turbine Release} \times \text{Timestep (hours)}}$$

17.1.1.12 Unit Power Table

This method uses a 3-D table that contains the columns Operating Head, Turbine Release, and Power for **each unit** in the plant.

Slots Added by this Method

Note, many of these slots have column or row dimensions based on the number of units. The rows/columns of these slots are expanded at the beginning of the run to match the value in the Number of Units slot. When first configuring this method, the user must enter the Number of Units, then run the model (stepping through 1 timestep is enough) to grow the slots to the right dimensions.

The following slots are instantiated when this method is selected.

☛ Auto Unit Best Turbine Q Table

Type: Table Slot

Units: LENGTH, FLOW
Description: Table showing most efficient release levels for given operating heads
Information: This table is generated from the Unit Power Table. The first column for each block is Operating Head and second column is Turbine Release. It will have one block for each unit.
I/O: Automatically generated at beginning of run
Links: Not Linkable

☛ AUTO UNIT MAX TURBINE Q TABLE

Type: Table Slot
Units: LENGTH, FLOW
Description: Table showing maximum release possible for given operating heads
Information: This table is generated from the Unit Power Table. The first column for each block is Operating Head and the second is Turbine Release. It will have one block for each unit.
I/O: Automatically generated
Links: Not Linkable

☛ MINIMUM POWER ELEVATION

Type: Table Slot
Units: LENGTH
Description: The minimum elevation at which the reservoir can still produce power.
Information:
I/O: Optional Input Only
Links: Not Linkable

☛ NUMBER OF UNITS

Type: Table Slot
Units: NONE
Description: Number of units in the plant
Information: This key scalar slot (existing slot, 1x1 table) indicates the number of units (turbines) at a power reservoir. The dimensions of several other slots are directly related to this value; specifically, both input data represented in Table Slots and unit-level series data represented in Agg Series Slots are require one row or column for each unit. At the **beginning of each run**, RiverWare will confirm that the value for the Number of Units slots is consistent with the dimensions of related slots. If any inconsistencies are detected, the relevant slots are resized as appropriate. If additional input data are required, the user is notified and the run is aborted.
I/O: Required Input only
Links: NA

☛ Number of Units Generating

Type: Series Slot
Units: NONE
Description: Number of units that are generating at a given timestep
Information: The value is the sum of the Unit Is Generating
I/O: Output only
Links: NA

☛ Power Curvature Tolerance

Type: Scalar Slot
Units: NONE
Description: The power curvature tolerance is used to account for anomalies in Unit Power Table data and round off error while calculating slopes.
Information: Although the units for the slot are “None”, the comparison is implicitly using (MW/cms)
I/O: Input or defaults to 1X10-6
Links: Links: Not linkable

☛ Unit Is Generating

Type: Agg Series Slot
Units: NONE
Description: This slot is used to control whether units are available.
Information: There is one column for each unit. Before a run, an input value of 1 indicates that the unit **must** generate power at that date; otherwise an input value of 0 indicates that the unit can **not** generate power at that timestep. A NaN indicates that the unit is available and the model will decide if it can generate. At the end of a run, an output 1 indicates the unit generated power, a 0 indicates it did not.
I/O: Can be input by user
Links: Not Linkable

☛ Unit Energy

Type: Agg Series Slot
Units: ENERGY
Description: Energy produced by each unit
Information: There is one column for each unit. A value indicates the energy being generated by the unit at that timestep, and takes into account frequency regulation. A negative value can be input or set by a rule to represent a unit that is spinning, motoring, or condensing (actually consuming energy).
I/O: Input, Rules, or Output
Links: Not linkable

☛ Unit Power

Type: Agg Series Slot
Units: Power
Description: The power that is generated by each unit
Information: There will be one column for each unit. A value indicates the power being generated by the unit at that timestep, and takes into account losses due to frequency regulation.
I/O: Calculated
Links: Not linkable

☛ Unit Power Table

Type: Table Slot
Units: Length, Flow, Power, Flow, Power, etc...
Description: A 3 dimensional table relating operating head, turbine release, and power for each unit in the plant. There will be 1 block (3 columns) for each unit.
Information: The last row for each operating head represents the **max capacity**. Best efficiency is automatically calculated. The three values in a given row and unit block represent a legal operating point for that unit, i.e., the Power which that unit would generate at that head and turbine flow. It will be necessary to enforce that a point of zero flow and zero power production be entered in the table for each operating head. It is also required that this table be concave.
I/O: Required Input only
Links: NA

Unit 1			Unit 2		
Operating Head (ft)	Turbine Flow (1000 cfs)	Power (KW)	Operating Head (ft)	Turbine Flow (1000 cfs)	Power (KW)
100	0	0	99	0	0
100	10	2000	99	10	1000
100	20	3000	99	20	2000
100	30	4000	99	30	3000
200	0	0	200	0	0
200	10	2500	200	10	1700
200	20	3500	200	20	2500
200	25	3800	200	25	2800
200	30	4500	200	30	3500
300	0	0	295	0	0
300	10	3000	295	10	3000
300	25	5000	295	25	4000

Unit Priority Table

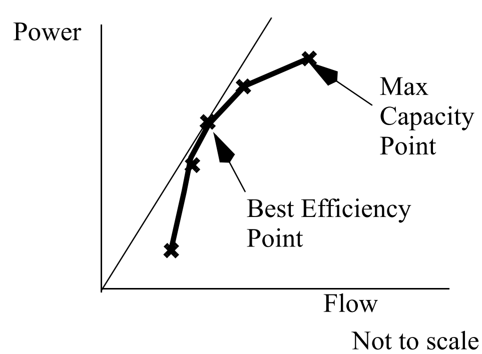
Type: Table Slot
Units: No Units
Description: The priority that each unit is started or stopped in the power plant
Information: There will be one row for each unit. In optimization only, units with lower numerical values are higher priority and are scheduled to release power in preference to lower priority units. If a value is absent then that unit is given the lowest priority. For units with equal priority, the unit efficiency will determine precedence, i.e. a unit with a higher efficiency will be prioritized higher than other units with the same priority value (note, not implemented yet. Currently, units with equal priority are turned on/off in an arbitrary order.). Currently, this table is only used in optimization. It is not used in simulation.
I/O: Optional input
Links: NA

Unit Turbine Release

Type: Agg Series Slot
Units: Flow
Description: Flow through each unit
Information: There is one column for each unit. The value is the expected Turbine Release through the unit at that timestep.
I/O: Input, Rules, Output
Links: Not Linkable

Method Details

This method will use table interpolation to calculate Power and Energy at a known Operating Head based on the characteristics of each unit. This method, Unit Power Table, works in a similar manner to the current method, Plant Efficiency Curve.



At the beginning of the run, the method creates the Auto Unit Best and Auto Unit Max Turbine Q tables from the Unit Power Table. The Auto Unit Max Turbine Q table are the points from the Unit Power Table that correspond to the largest Turbine Release for a given Operating Head. The Auto Unit Best Turbine Q table is calculated from the Unit Power Table as follows: For each unit and each operating head, the method determines the point (flow, power) that corresponds to a line drawn from the origin and it tangent to the curve. This tangent point is determined by calculating the slope of the line from each point to the origin; the point with the largest slope is the tangent. Then the operating head and turbine flow for this point are added to the Auto Unit Best Turbine Q.

The description of the solution in this section assumes that mass balance has already occurred (i.e. Inflow, Outflow, Storage, Pool Elevation have been calculated) and the method is trying to compute the Energy and Power produced by that Outflow. At the start of this method, there is also an estimate of the Turbine Release calculated as

$$\text{Turbine Release} = \text{Outflow} - (\text{Unregulated Spill} + \text{Regulated Spill} + \text{Bypass})$$

where the spills are either known or estimated based on the current Pool Elevation. This Turbine Release may be reset if it cannot be met. If Energy is input, set by a rule, or flagged **BEST EFFICIENCY** (B) or **MAX CAPACITY** (M) or **UNIT** (U) then Energy and Power are solved using the method **Unit Power Table Release** described in Section 17.1.2.8 on page 439.

In this description, "t" indicates the current timestep and "u" indicates that the method will do this for each unit.

The method calculates the tailwater and operating head using the selected method and current release and pool elevation.

For each unit, the previous pool elevation will be compared to the unit's minimum power elevation.

```
if (Pool Elevation[t-1] < Unit Minimum Power Elevation[u])
{
    {
    Either set the following to zero or make sure that they are zero:
    • Unit Turbine Release[t,u],
    • Unit Energy[t,u],
    • Unit Power[t,u], and
    • Unit is Generating[t,u].
    If any of these are non-zero, an error is issued.
    }
}
```

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Power, Energy, and Turbine Release are set equal to zero.

2. Either set the following to zero or make sure that they are zero:
   - Unit Turbine Release[t,u],
   - Unit Energy[t,u],
   - Unit Power[t,u], and
   - Unit is Generating[t,u].

   If any of these are non-zero, an error is issued No further computations are performed.

Next for each unit, an estimate of max flow through all the turbines is calculated as follows: estimate a temporary variable maxPowerRelease (flow) using the Auto Unit Max Turbine Q table. This table contains the columns Operating Head and Turbine Capacity. Because Operating Head is known at the current timestep, table interpolation is used to calculate maxPowerRelease for the given average Operating Head.

If Outflow is set to **Max Capacity** flag, set the Unit Turbine Release to the maximum calculated and compute the power produced by those flows.

Otherwise, if Unit Turbine Release is input/rules for any of the units:

If Unit is Generating is set (input/rules) to 0 for a unit that has a Unit Turbine release, issue an error.
If Unit is Generating is set (input/rules) to 1 for a unit that does not have a Unit Turbine release, issue an error.
If Turbine Release is not set by the U flag, check if Turbine Release = Σ Unit Turbine Release[u]. If they do not match and Turbine Release is input/rules, issue an error. Otherwise, if they do not match, the method resets Turbine Release equal to Σ Unit Turbine Release[u].
If Turbine Release does have the U flag, the Turbine Release is set equal to Σ Unit Turbine Release[u].
If Turbine Release is now greater than Σ maxPowerRelease[u], an error is issued as the specified unit turbine releases cannot be met.
If a regulation method is selected, call it here, otherwise, given the known Unit Turbine Releases[u], the method then looks up the unit flow and operating head on the Unit Power Table to determine the power produced by each unit: Unit Power[u].
Finally, Unit Energy[u] = Unit Power[u] * Time (hrs)
The total Power = Σ Unit Power[u] and total Energy = Σ Unit Energy [u]

Else Unit Turbine Release is not input/rules

Exit the method as there is no way to compute energy/power at a unit level. The dispatch method will continue but no power related slots (e.g. Energy, Unit Energy, Power, Unit Power, Unit Turbine Release, Unit Is Generating, Unit Startup, Unit Shutdown, etc) will be set. Turbine Release is set to the minimum of (Outflow - Spill or MaxPowerRelease).

This can happen when dispatching given Inflow and Pool Elevation or Storage. and no Turbine Release or Unit Turbine Release is specified.

Finally, the method computes the slot Unit is Generating based on the Unit Turbine Release and Unit Energy. For each unit, if these are non zero, the Unit is Generating is set to 1. If they are zero, Unit is Generating is set to 0. No inputs are overwritten. Then, the Number of Units Generating is computed as the sum over the columns of the Unit is Generating slot.

17.1.2 Power Release

When Energy is specified, the Power Release method is used to calculate the Turbine Release required. If the Energy request can not be met, the user is notified. There is one method per Power method.

The Power Release category is available when any of the Power methods is selected **except** None, Peak Power, Peak and Base, or Peak Power Equation.

17.1.2.1 None

This is the default method in the Power Release category. No calculations are performed in this method. There are no slots specifically associated with this method. If this method is selected for the Power Release category, a **RiverWare**™ error will be posted and the simulation run will be aborted. A viable power release method must be selected when the Power Release category is visible.

SLOTS SPECIFIC TO THIS METHOD

☛ **NONE**

17.1.2.2 Plant Power Coefficient Release

The Plant Power Coefficient Release method calculates Turbine Release using the entire plant characteristics when Energy is specified. The Plant Power Coefficient Release method is only available if the Plant Power Coefficient method is selected in the Power category. Energy must be input for this method to execute. If Energy is flagged as either **MAX CAPACITY** or **BEST EFFICIENCY**, it is considered input. If Energy is flagged as **MAX CAPACITY**, Turbine Release is set to meet the Energy request at the maximum flow rate. If Energy is flagged as **BEST EFFICIENCY**, Turbine Release is set to meet the Energy request at the most efficient flow rate. If Energy is neither flagged as **MAX CAPACITY** nor flagged as **BEST EFFICIENCY**, the Turbine Release is calculated from the Energy request and a Power Coefficient. The Power Coefficient may be input by the user or calculated by **RiverWare**™ from interpolation of the Best and Max Power Coefficient tables.

If Energy is flagged **UNIT VALUES** (U), and error is issued. This flag is only available with the “Unit Power Table Release” method.

SLOTS SPECIFIC TO THIS METHOD

☛ **NONE**

The first step in the Plant Power Coefficient Release algorithm is to set the Power Plant Cap Fraction to 1.0 if it is not already known.

If the Energy slot is flagged as MAX CAPACITY, the following steps are taken:

1. Qtemp, a local variable, is calculated from interpolation of the Max Turbine Q table using the Operating Head.

2. Turbine Release is calculated with the following equation:

$$\text{Turbine Release} = \text{Qtemp} \times \text{Power Plant Cap Fraction}$$

3. PCmax, a local variable, is determined from interpolation of the Max Power Coefficient table using the Operating Head. The Power Coefficient is set as PCmax if it is not input.

4. Power and Energy are then calculated using the following equations:

$$\text{Power} = \text{Turbine Release} \times \text{PCmax}$$

$$\text{Energy} = \text{Power} \times \text{Timestep (in hours)}$$

5. If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit and the Energy is recalculated. A new Power Coefficient and Turbine Release are then calculated based on the Plant Power Limit.

If Energy is flagged as BEST EFFICIENCY, the following steps are taken:

1. Qtemp is calculated from interpolation of the Best Turbine Q table using the Operating Head.

2. Turbine Release is computed using the following equation:

$$\text{Turbine Release} = \text{Qtemp} \times \text{Power Plant Cap Fraction}$$

3. PCbest, a local variable, is determined from interpolation of the Best Power Coefficient table using the Operating Head. The Power Coefficient is set as PC best if it is not input.

4. Power and Energy are then calculated using the following equations:

$$\text{Power} = \text{Turbine Release} \times \text{PCbest}$$

$$\text{Energy} = \text{Power} - \text{Timestep (in hours)}$$

5. If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit, the Energy is recalculated, and the Turbine Release is recalculated as Plant Power Limit / PCbest.

If Energy is not flagged as either MAX CAPACITY or BEST EFFICIENCY and the Power Coefficient is input, the following steps are taken.

1. If the Power Coefficient is less than 0.00000001, a **RiverWare**™ error is posted and the simulation run is aborted.
2. Power is calculated using the following equation:

$$\text{Power} = \text{Energy} / \text{Timestep}$$

where the Timestep is in hours.

3. Qout, a local variable, is calculated with the following equation:

$$\text{Qout} = \text{Power} / \text{Power Coefficient}$$

4. Qtemp, a local variable, is determined by the interpolation of the Max Turbine Q table using the Operating Head.
5. Qmax, a local variable, is computed using the following equation:

$$\text{Qmax} = \text{Qtemp} \times \text{Power Plant Cap Fraction}$$

6. If Qout is greater than Qmax, the largest discharge value in the Max Turbine Q table is found. If this value is greater than or equal to Qout, Turbine Release is set equal to Qmax. If the value is less than Qout, a **RiverWare**™ error is posted and the simulation run is aborted.
7. If Qout is less than or equal to Qmax, Turbine Release is set equal to Qout.

If Energy is not flagged as either MAX CAPACITY or BEST EFFICIENCY and the Power Coefficient is not given, the following steps are taken:

1. Power is calculated using the following equation:

$$\text{Power} = \text{Energy} / \text{Timestep}$$

where the timestep is in hours.

2. The best and max power coefficients are interpolated using the Operating Head and the Best Power Coefficient and the Max Power Coefficient tables, respectively.
3. QbestTemp and QmaxTemp (local variables) are then determined using the Operating Head to interpolate values from the Best Turbine Q and Max Turbine Q tables, respectively.

4. Qbest, a local variable, is computed using the following equation:

$$\text{Qbest} = \text{QbestTemp} \times \text{Power Plant Cap Fraction}$$

5. Qmax, a local variable, is calculated using the following equation:

$$\text{Qmax} = \text{QmaxTemp} \times \text{Power Plant Cap Fraction}$$

6. If Power divided by the best power coefficient is less than or equal to Qbest, Turbine Release is set equal to Power divided by the best power coefficient.

7. If Power divided by the max power coefficient is greater than Qmax, Turbine Release is set equal to the max turbine flow.

8. If neither 3) nor 4) is true, an interpolated value (pcoeffINTERP) is found between the best and max power coefficients based on how close Power is to both the product of Qbest and the best power coefficient, and the product of Qmax and the max power coefficient. The following pair of equations is used to quantitatively determine the pcoeffINTERP value:

$$\text{pcoeffFRACTION} = \frac{(\text{Power} - \text{best power coefficient} \times \text{Qbest})}{(\text{max power coefficient} \times \text{Qmax} - \text{best power coefficient} \times \text{Qbest})}$$

$$\text{pcoeffINTERP} = \text{best power coefficient} + (\text{max power coefficient} - \text{best power coefficient}) \times \text{pcoeffFRACTION}$$

9. The Turbine Release is then calculated with the following equation:

$$\text{Turbine Release} = \text{Power} / \text{pcoeffINTERP}$$

17.1.2.3 Plant Efficiency Curve Release

The Plant Efficiency Curve Release method calculates Turbine Release using the entire plant characteristics when Energy is specified. The Plant Efficiency Curve Release method is only available if the Plant Efficiency Curve method is selected in the Power category. Energy must be input or set by a rule for this method to execute. If Energy is flagged as either **MAX CAPACITY** or **BEST EFFICIENCY**, it is considered input. If Energy is flagged as **MAX CAPACITY**, Turbine Release is set to meet the Energy request at the maximum flow rate. If Energy is flagged as **BEST EFFICIENCY**, Turbine Release is set to meet the Energy request at the most efficient flow rate. If Energy is neither flagged as **MAX CAPACITY** nor flagged as **BEST EFFICIENCY**, the Turbine Release is calculated from the Energy request.

If Energy is flagged **UNIT VALUES** (U), and error is issued. This flag is only available with the “Unit Power Table Release” method.

Slots Specific to this Method

☛ NONE

The first step in the Plant Efficiency Curve Release algorithm is to set the Power Plant Cap Fraction to 1.0 if it is not already known.

If the Energy slot is flagged as MAX CAPACITY, the following steps are taken:

1. Qtemp, a local variable, is calculated as the maximum release using the Operating Head and the Plant Power Table.
2. Turbine Release is calculated with the following equation:

$$\text{Turbine Release} = \text{Qtemp} \times \text{Power Plant Cap Fraction}$$

3. Power is determined directly from the Plant Power Curve.
4. Energy is calculated as:

$$\text{Energy} = \text{Power} \times \text{Timestep}$$

5. The Power Coefficient is calculated as:

$$PowerCoefficient = (Power)/(TurbineRelease)$$

6. If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit and the Energy is recalculated. A new Power Coefficient and Turbine Release are then calculated based on the Plant Power Limit.

If Energy is flagged as BEST EFFICIENCY, the following steps are taken:

1. Qtemp is computed as the most efficient release given the Operating Head and the Plant Power Table.
2. Turbine Release is computed using the following equation:

$$\text{Turbine Release} = \text{Qtemp} \times \text{Power Plant Cap Fraction}$$

3. Power is determined directly from the Plant Power Curve.
4. Energy is calculated as:

$$\text{Energy} = \text{Power} \times \text{Timestep}$$

5. The Power Coefficient is calculated as:

$$PowerCoefficient = (Power)/(TurbineRelease)$$

6. If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit and the Energy is recalculated. A new Power Coefficient and Turbine Release are then calculated based on the Plant Power Limit.

If Energy is not flagged as either MAX CAPACITY or BEST EFFICIENCY and the Power Coefficient is input, the following steps are taken.

1. If the Power Coefficient is less than 0.00000001, a **RiverWare**™ error is posted and the simulation run is aborted.
2. Power is calculated using the following equation:

$$\text{Power} = \text{Energy}/\text{Timestep}$$

3. Turbine Release is calculated as:

$$\text{Turbine Release} = \text{Power}/\text{Power Coefficient}$$

If Energy is not flagged as either MAX CAPACITY or BEST EFFICIENCY and the Power Coefficient is not input, the following steps are taken:

1. Power is calculated using the following equation:

$$\text{Power} = \text{Energy}/\text{Timestep}$$

2. The max Turbine Release and Power production are found for the current operating conditions.
3. If input Power is greater than the max Power for current operating conditions, and INPUT_ENERGY _ADJUST method is chosen, Turbine Release is set equal to the max Turbine Release from 2, and Power is set equal to Power from 2. The Power Coefficient is then computed as Power divided by Turbine Release.
4. Otherwise, Turbine Release is found using the Plant Power Table and the Power Coefficient is set as Power divided by Turbine Release.
5. If the Plant Power Limit is exceeded, an error is posted.

Notes on Power Plant Cap Fraction

If the Power Plant Cap Fraction is input by the user, it is necessary for the Plant Power Table to basically be scaled back to account for the operating points when the turbines are operating at less than 100%. To do this, when Turbine Release is known and Power is to be found using the Plant Power Curve, Turbine Release is divided by the Power Plant Cap Fraction. This point is then found in the Plant Power Curve for the current operating head and the Power is found using 3-D interpolation. Finally the Power is multiplied by the Power Plant Cap Fraction to get the actual Power produced for the current timestep.

If Power is known, and Turbine release is to be found in the table. Power is multiplied by the Power Plant Cap Fraction and then this point is found in the Plant Power Curve to solve for Turbine Release. Turbine Release is then divided by the Power Plant Cap Fraction to get the actual Turbine Release for the current timestep.

17.1.2.4 Plant Power Equation Release

The Plant Power Equation Release method calculates Turbine Release using the water power equation when Energy is specified. The Plant Power Equation Release method is only available if the Plant Power Equation method is selected in the Power category. Energy must be input for this method to execute. If Energy is flagged as either **MAX CAPACITY** or **BEST EFFICIENCY**, it is considered input. If Energy is flagged as **MAX CAPACITY**, Turbine Release is set to meet the Energy request at the maximum possible turbine release. If Energy is flagged as **BEST EFFICIENCY**, the run aborts because **BEST EFFICIENCY** is not supported in this method.

If Energy is flagged **UNIT VALUES** (U), and error is issued. This flag is only available with the “Unit Power Table Release” method.

Slots Specific to this Method

☛ NONE

This method first checks to see if Turbine Release is user input or set by a rule. If it is, the run aborts because both Energy and Turbine Release cannot be input.

If the Energy slot is flagged as MAX CAPACITY, the following steps are taken:

1. Set Turbine Release to be the maximum turbine release calculated by interpolating the Net Head on the Net Head Vs Max Turbine Release table.

2. Once efficiency, Plant Cap Fraction, Net Head, and Turbine Release are all known, Power is solved for using the Power Equation. The unit compatibility factor comes from balancing units and is 102.01697767 in internal RiverWare units.

$$Power = \frac{\text{Turbine Release} \times \text{Net Head} \times \text{efficiency} \times \text{Plant Cap Fraction}}{\text{Unit Compatibility Factor}}$$

If the computed Power is greater than the Plant Power Limit, the Power is reset to the Plant Power Limit. In this case, Turbine Release is re-computed using the above equation rearranged.

3. Lastly, Energy is computed as Power multiplied by the length of the timestep.

$$Energy = \text{Power} \times \text{Length of Timestep}$$

If the Energy slot is not flagged MAX CAPACITY, the following steps are taken:

When the Energy value is known (rather than flagged **Max Capacity**), the Plant Power Equation Release method uses Energy to solve for Power and Turbine Release. Power is simply Energy divided by the length of the timestep:

$$Power = \frac{Energy}{\text{Length of Timestep}}$$

Using Power, the Net Head and Turbine Release are solved for iteratively as described below:

1. If the computed Power is greater than the Plant Power Limit, the specified energy is too large. The selected method in the Input Energy adjustment category is executed. The Reduce Input Energy method reduces the energy to the maximum possible. If the None method is selected, an error will be issued that the specified energy leads to a power that is above the Plant Power Limit.

2. Turbine Release is initially assumed zero

3. Tailwater Elevation is determined via the selected Tailwater method (the "flow" variable is set to Outflow. If Turbine Release is linked it can be assumed that the Turbine Release and Spill are separated and the "flow" variable should be set to Turbine Release.)

4. Operating Head is calculated as Pool Elevation minus Tailwater Elevation

5. Net Head is calculated as Operating Head minus Head Loss

6. Turbine Release is calculated again using the Water Power equation:

$$TurbineRelease = \frac{Power \times \text{Unit Compatibility Factor}}{\text{Net Head} \times \text{Plant Efficiency}}$$

7. The calculated Turbine Release is compared to the initial Turbine Release and the process iterates until the values converge. (Note: Convergence Percentage is a general slot on power reservoirs representing the convergence in all iterative solutions-- the slot defaults to 0.0001 if not user input.)

Once converged, the Net Head is looked up on the Net Head Vs Max Turbine Release table to get the max release. If the Turbine Release is larger than the max release times the Power Plant Cap Fraction, the selected method in the Input Energy adjustment category is executed. The Reduce Input Energy method reduces the energy to the maximum possible. Otherwise, there is too much flow and an error will be issued that the energy request cannot be met.

17.1.2.5 Peak Power Equation with Off Peak Spill Release

The Peak Power Equation with Off Peak Spill Release method calculates the necessary Turbine Release, Peak Release and Peak Time using the water power equation when Energy is specified. The method is only available if the Peak Power Equation with Off Peak Spill method is selected in the Power category. Energy must be input or set by a rule for this method to execute.

SLOTS SPECIFIC TO THIS METHOD

☛ NONE

This method first checks to see if Turbine Release is user input or set by a rule. If it is, the run aborts because both Energy and Turbine Release cannot be input. When the Energy value is known, the Peak Power Equation with Off Peak Spill Release method uses Energy to solve for Turbine Release, Peak Release and Peak Time as described below:

1. Peak Release is initially set to zero.
2. Given the net head from the previous timestep (Operating Head at previous timestep minus Head Loss), the efficiency is interpolated from the Net Head vs Efficiency table. The previous Operating Head is used as an approximation so as not to introduce an additional variable in the iteration. As a result, the Tailwater Elevation at the initial timestep must be input. The net head for the initial timestep is the initial Pool Elevation minus the initial Tailwater Elevation minus Head Loss.
3. The current Tailwater Elevation is determined using the maximum of Peak Release or the current Outflow as the value in the selected Tailwater method.
4. The Operating Head is calculated as the average Pool Elevation minus the Tailwater Elevation.
5. The net head is calculated as the Operating Head minus the Head Loss.
6. Given the net head, the Generator Capacity is interpolated from the Net Head vs. Generator Capacity table. If the capacity is above the Plant Power Limit, the Generator Capacity is reset to the Plant Power Limit.
7. Peak Release is calculated according to the power equation. The unit compatibility factor comes from balancing units and the specific weight of water; it is 102.01697767 in internal RiverWare units.

$$\text{Peak Release} = \frac{\text{Generator Capacity} \times \text{Unit Compatibility Factor}}{\text{Net Head} \times \text{Efficiency}}$$

8. The new Peak Release value is compared with the previous value and the iteration (steps 3-7) continues until the value converges. (Note: Convergence Percentage is a

general slot on power reservoirs representing the convergence in all iterative solutions-- the slot defaults to 0.0001 if not input.)

Power is set equal to the Generator Capacity and Peak Time is:

$$\text{Peak Time} = \frac{Energy}{Power}$$

Turbine Release is the Peak Release average over the timestep:

$$\text{Turbine Release} = \frac{\text{Peak Release} \times \text{Peak Time}}{\text{Timestep Length}}$$

17.1.2.6 Unit Generator Power Release

The Unit Generator Power Release method is only available when Unit Generator Power is selected in the Power category. It is used to calculate the Turbine Release required to produce a given amount of Power. Energy must be input by the user for this method to execute. There are no slots specifically associated with this method.

If Energy is flagged **UNIT VALUES** (U), and error is issued. This flag is only available with the “Unit Power Table Release” method.

The Unit Generator Power Release method begins by computing the availability and power limits of each unit type. Availability and power limit values are computed as the sum of the values from the availability and power limit columns, respectively, in the Generators Available and Limit slot. A value for availability and power limit is computed for each unit type.

The efficiency of each unit type is calculated by the following equation:

$$efficiency = \frac{powerTemp}{flowTemp}$$

PowerTemp and flowTemp, both local variables, are computed from the Best Generator Power and Best Generator Flow tables, respectively, using the current Operating Head. Each unit type is then sorted in descending order based on the computed efficiency.

In order to compute the Turbine Release associated with the known Power, the method begins to add entire unit types (operating according to the best power and flow tables and beginning with the most efficient type) until the Power is exceed or all the unit types have been added. If the Power is exceeded, the last generator type is interpolated to compute the Turbine Release exactly (see equation below). However, if all the unit types have been added and the Power cannot be met, the method assumes all unit types are operating at full capacity (according to the Full Generator Flow and Full Generator Power tables). Then if the Power is exceeded, the last generator type added is interpolated to compute the Power exactly (see equation below). However, if the Power still cannot be met, an error is posted and the run is aborted because the generators are unable to produce the amount of Power specified by the user.

The interpolation equation used to calculate Power is given below:

$$\text{Turbine Release} = \text{oneLessTypeFlow} + \frac{\text{Power} - \text{oneLessTypePower}}{\text{cumulativePower} - \text{oneLessTypePower}} \bullet (\text{cumulativeFlow} - \text{oneLessTypeFlow})$$

where oneLessTypePower is the power produced from all the previous types added (excluding the most recent type added); oneLessTypeFlow is the flow through all the previous unit types (excluding the most recent type added); cumulativePower is the power produced from all the unit types added (including the most recent type); and cumulativeFlow is the flow through all the unit types added (including the most recent type).

Note: The above equation assumes the relationship between power and flow is linear regardless of the actual relationship specified in the power and flow tables. It is also interpolating over an entire type of generators.

17.1.2.7 LCR Power Release

The LCR Power Release method calculates the release from the Lower Colorado River hydropower products. The LCR Power Release method is available only when LCR Power is selected in the Power category. Energy must be input or flagged as **BEST EFFICIENCY** (Energy cannot be flagged **MAX CAPACITY** for the LCR Power method) for this method to execute. It is determined if the requested Power demand can be met. This determination is based on the maximum possible power that can be generated for a given head. If it is possible to meet the requested Power demand, the Turbine Release is set so as to produce the requested Power.

If Energy is flagged **UNIT VALUES** (U), and error is issued. This flag is only available with the “Unit Power Table Release” method.

Slots Specific to this Method

☛ None

The first step in this method is making sure the Lower Colo Power Coeffs are known. If either of these coefficients are not known, a **RiverWare**™ error is flagged and the simulation run is aborted. Then, the LCR Input Efficiency slot is checked. is not known, If it is not known, it is assumed to be 100% efficient and the LCR Input Efficiency is set to 1.0.

If Energy is flagged as **BEST EFFICIENCY**, it is calculated as the Net Energy Request plus the value of energy in the Station Energy Table corresponding to the current day of the week.

If Energy is flagged as **MAX CAPACITY**, an error is given. If Energy is not flagged as either **BEST EFFICIENCY** or **MAX CAPACITY**, it must be input by the user.

Turbine Release is calculated using the following equation:

$$\text{Turbine Release} = \left(\frac{\text{Energy}}{\text{LCR Input Efficiency}} + (1000 \times \text{Lower Colo Power Coeff \#2})\right)$$
$$\times \frac{1000}{\text{Lower Colo Power Coeff \#1} \times \text{Timestep} \times 62.4/737.5 \times \text{Operating Head} \times 35.31467 \times 3.28084}$$

where the Timestep is in hours. The constants used in the above equation are to convert the input to **RiverWare**™ standard units.

The previous equation is based on the energy calculation equation solved for Flow and corrected to standard units (see LCR Power method):

$$\text{Energy (1000 MWH)} = \left(\left(\text{Lower Colo Power Coeff \#1} \times \frac{62.4}{737.5} \times \text{Flow (Kcfs)}\right.\right.$$
$$\left.\times \text{Timestep (hours)} \times \frac{\text{Operating Head (ft)}}{1000}\right)$$
$$\left. - \text{Lower Colo Power Coeff \#2}\right) \times \text{LCR Input Efficiency}$$

where flow is in kcfs, Timestep is in hour, and Operating Head is in feet.

The correction factors used in the above equations are presented below:

$$\text{Energy (1000 MWH)} = \text{Energy (MWH)} \times 1000$$

$$\text{Flow (cfs)} = \text{Flow (Kcfs)}/1000$$

$$\text{Operating Head (m)} = \text{Operating Head (ft)}/3.28084$$

$$\text{Turbine Release (cms)} = \text{Flow (cfs)}/35.31467$$

Once Turbine Release is calculated, it is checked against the maximum allowable turbine release. A **RiverWare**™ error is flagged and the simulation run is aborted if Turbine Release exceeds the maximum allowable turbine release.

17.1.2.8 Unit Power Table Release

This method is only available if the Unit Power Table method is selected in the Power category, (See “Unit Power Table Release” on page 439.). The method **Unit Power Table Release** calculates Turbine Release when Energy is specified. If Energy is flagged as **BEST EFFICIENCY** (B) or **MAX CAPACITY** (M) or **UNIT VALUES** (U), it is considered input.

If Energy is flagged B, the Unit Best Turbine Q table will be used to determine the best efficiency Turbine Release for the current average Operating Head. This assumes that all units are in use unless specified otherwise in the Unit is Generating slot. Power is then found using the Unit Power Table. If Energy is flagged M, the Unit Max Turbine Q table is used to determine the maximum Unit Turbine Release for the current average Operating Head. This point is then found in the Unit Power Table to determine the maximum power that can be produced for this Operating Head. If Energy is flagged U, the method calculates Unit Turbine Release using table interpolation of Unit Energy on the Unit Power Table with the Unit Energy.

If Energy is input but not flagged as B, M, or U and Unit Energy is not input, the method will exit without calculating Unit Energy. If any of the values in Unit Energy are input, it will be used to determine the release and power.

Method Details

This method will be called if Energy is input or set by a rule, which includes being flagged B, M, or U. This method will execute in the following manner.

```
if (Energy is flagged M)
{
    If any of the Unit Energy[u] values are input or set by a rule, issue an error.
    For each unit that is available (based on a non-zero value in the Unit is Generating
    slot), use 2D interpolation of Auto Unit Max Turbine Q table;
    Set max release to a temporary local variable, Qmax[u];
    Turbine Release is set to Σ Qmax[u];
    Once the value for each unit flow at the current average Operating Head is found, the
    Unit Power[u] produced for that flow can be determined directly from the Unit Power
    Table.
}
else if (Energy is flagged B)
{
    If any of the Unit Energy[u] values are input or set by a rule, issue an error.
    For each unit that is available (based on a non-zero value in the Unit is Generating
    slot), use 2D interpolation of Auto Unit Best Turbine Q table to determine release at B;
    Set best release to a temporary local variable, Qbest[u];
    Turbine Release is set to Σ Qbest[u];
    Again, Unit Power[u] will then be able to be determined directly from the Unit Power
    Table.
}
else if (Energy is Input/Rules (including U flag) and Unit Energy for any unit is not input/
rules)
{
    Issue an error; there is no way to calculate Unit Energy from plant values and no way
    to calculate plant Power without unit information
```

}
else if (Energy is input/rules (including U flag) and Unit Energy for any unit is input/rules)
{
If Unit is Generating is set (input/rules) to 0 for a unit that has a Unit Energy, issue an error.
If Unit is Generating is set (input/rules) to 1 for a unit that does not have a Unit Energy, issue an error.
If Energy is flagged U, $Energy = \Sigma UnitEnergy[u]$; otherwise, if $Energy \neq \Sigma UnitEnergy[u]$, an error is issued
Next, Unit Power[u] = Unit Energy[u] / time (hrs)
$Power = \Sigma UnitPower[u]$
From this power calculation, the Unit Turbine Release[u] can then be determined using a reverse table lookup of Unit Power[u] in the Unit Power Table. If the Shared Penstock Head Loss method is selected, the solution is iterative as the net operating head is a function of Turbine Release. If Unit Energy[u] is less than zero, the Unit Turbine Release[u] is set to zero. A negative Unit Energy can be set to represent a unit that is spinning but not producing power (i.e. condensing).
Turbine Release = Σ Unit Turbine Release[u]
}
Finally, the method returns to the Unit Power Table method and computes Unit is Generating and Number of Units Generating. See “Unit Power Table” on page 420..

17.1.3 Power Unit Information

This category is used to provide information on unit information when one of two plant level power methods is selected. This category is only available if the **Plant Efficiency Curve** or **Plant Power Coefficient** methods are selected.. In this category are two methods:

17.1.3.1 None

This is the default, no-action method.

17.1.3.2 Plant Power Table with Units

When selected, the **Plant Power Table with Units** method allows the user to specify the number of units associated with each Turbine Release / Power combination on the Plant Power Table. In addition, at the end of the power method, the Operating Head and Turbine Release are looked up to compute the number of units that are generating.

SLOTS ASSOCIATED WITH THE METHOD:

☛ **PLANT POWER TABLE**

Type: Table Slot
Units: LENGTH, FLOW, POWER, NONE
Description: 3-D table used to determine power using interpolation
Information: Data must be entered into the table in increasing, concave blocks of the same Operating Head for the 3-dimensional table interpolation to work correctly. For every block of the same Operating Head in column 1, Turbine Release should be listed in increasing, concave order in column 2, and the corresponding Power in column 3. The number of units should be increasing in column 4. There should also be a point of zero Turbine Release and zero Power for each operating head. The second to last row for each operating head is the point of best efficiency. The last row for each operating head is the point of maximum Turbine Release and maximum Power production. If there are only two rows for a given operating head, both the **best efficiency** and **max capacity** are equal to the second row. The table shown below is an example of the proper way to formulate the Plant Power Table, with units.

Operating Head	Turbine Release	Power	Number of Units
30	0	0	0
30	100	100	1
30	200	175	2
40	0	0	0
40	100	125	1
40	220	195	2

Operating Head	Turbine Release	Power	Number of Units
50	0	0	0
50	110	147	1
50	250	205	2

I/O: Input Only
Links: Not Linkable

☛ Number of Units Generating

Type: Series
Units: NONE
Description: The number of units generating on this timestep
Information: The value in this slot is computed by looking up the Operating Head and Turbine Release on the Plant Power Table to find the number of units. Note, this computation is a 3D interpolation on the Plant Power Table so there can be a fractional number of units generating, i.e. 1.7.
I/O: Output Only
Links: Not linkable

17.1.4 Input Energy Adjustment

This method category is only available for Plant Power Coefficient Release, Plant Efficiency Curve Release, or Plant Power Equation Release methods in the Power Release category. Its purpose is to adjust input Energy values if they violate a physical constraint.

17.1.4.1 None

This is the default method. It performs no calculations and there are no slots associated with it. The Energy values will not be adjusted if this method is selected.

17.1.4.2 Reduce Input Energy

This method is used to reduce the input Energy value whenever it exceeds the maximum power (due to turbine capacity).

Slots Specific to this Method

☛ **Requested Energy**

Type: Series Slot
Units: FLOW
Description: The Energy value before being adjusted
Information: This slot is available so that the user can see when an Energy value is adjusted. The value in this slot is the energy value before being adjusted. A value exists in this slot only if the Energy value is adjusted.
I/O: Output only
Links: Not linkable

If the Energy slot value leads to a power that is greater than the maximum reservoir power (due to plant capacity, Plant Power Limit, etc), this method saves the Energy value in the Requested Energy slot. Then, the Maximum Capacity flag is set on the Energy slot. The reservoir is then forced to resolve with the Energy set to Max Capacity (instead of the original, input value). When the reservoir solves the second time, it computes the maximum reservoir Energy and sets this value on the Energy slot. The Maximum Capacity flag remains on the Energy slot for the timestep in question (and will be saved with the model file).

17.1.5 Power Plant Failure

This category is available when any of the power methods are chosen.

17.1.5.1 None

No power plant failure is modeled. If not input, the **Power Plant Cap Fraction**, if used, is set to 1.0.

17.1.5.2 Max Pool, Outflow, Tailwater

During high flow events, certain conditions cause the power plant to fail and no power can be produced. This method model the following conditions:

- Maximum pool elevation
- Maximum tailwater elevation
- Maximum outflow

Each of the above actually has two values, the first, lower value represents the shutoff criteria. The second, higher value represents the failure criteria. If the shutoff criteria is exceeded, then no power can be produced for that timestep, but if the conditions receded below the criteria, then power can again be produced. If the failure criteria is exceeded, the power plant has failed and no power can be produced from that point forward.

For example, a Pool Elevation above the shutoff limit requires the plant to cease generation. However, the power house is not flooded, and when the pool drops back below this limit, the plant can resume generation. In the second case, the pool is above the failure limit and requires the plant to cease generation, but also floods the powerhouse. In this case, even if the pool drops back below the criteria, the plant cannot resume generation, i.e the plant must “fail” for the rest of the simulation.

SLOTS ADDED BY THIS METHOD:

This method will instantiate slots in the following list.

☛ MAX POOL ELEVATION FOR POWER

Type: Table
Units: LENGTH, LENGTH
Description: Elevations at which no power can be produced
Information: This is 1X2 table slot. The first column contains the **shutoff** elevation. The second column contains a higher elevation representing the **failure** elevation.
I/O: Optional input
Links: NA

☛ MAX TAILWATER ELEVATION FOR POWER

Type: Table

Units: LENGTH, LENGTH
Description: Tailwater elevation at which power can no longer be generated
Information: This is 1X2 table slot. The first column contains the **shutoff** tailwater elevation. The second column contains a higher elevation representing the **failure** tailwater elevation. Once the failure elevation is exceeded, the plant has failed and no power can be produced on any subsequent timesteps.
I/O: Optional input
Links: NA

☛ MAX OUTFLOW FOR POWER

Type: Table
Units: FLOW, FLOW
Description: Reservoir outflow at which power can no longer be generated
Information: This is 1X2 table slot. The first column contains the **shutoff** outflow. The second column contains a higher outflow representing the outflow at which the power plant **fails**.
I/O: Optional input
Links: NA

☛ POWER PLANT CAP FRACTION

Type: Series
Units: NO UNITS
Description: This slot tracks whether power production is possible at this timestep
Information: A value of 1 indicates power can be generated, a value of 0 indicates no power can be generated.
I/O: Input or Output
Links: NA

METHOD DETAILS:

This method is executed at the beginning of each power method. First, the failure conditions are checked, then the shutoff conditions are checked.

Failure is tracked using the **Power Plant Cap Fraction.** If **Power Plant Cap Fraction** is 1.0 the power plant is available. If it is 0.0, the plant has failed.

At the beginning of the power method, the following logic determines the **Power Plant Cap Fraction** to use.

> If the **Power Plant Cap Fraction** is input, that input value is used and no further checking will be done.
> If the previous **Power Plant Cap Fraction** is 0.0, then the current **Power Plant Cap Fraction** is set to zero. This indicates the power plant has failed on previous timesteps and should remain failed. No further checking is done.
> Else, the previous **Power Plant Cap Fraction** is unknown or non-zero. Then, if any of the

following are true, then the **Power Plant Cap Fraction** is set to 0.0; the plant has failed.

$$\text{Pool Elevation}[t-1] > Max \text{ Elevation for Power}[failure]$$

$$\text{Tailwater Elevation}[t-1] > \text{Max TailwaterElevation for Power}[failure]$$

$$Outflow[t-1] > Max \text{ Outflow for Power}[failure]$$

A diagnostic is available in the User Methods category that describes any failure constraints in effect.
Otherwise, the **Power Plant Cap Fraction** is set to 1.0
Even if the plant "fails" in the course of a simulation, the user can "restart" it manually by setting a non-zero value in the **Power Plant Cap Fraction.**

If any of the following are true, then the plant has failed or shutoff. Turbine Release, Power, Energy and a few method specific slots (as described in the power method section) are set to zero.

$$\text{Power Plant Cap Fraction}[t] = 0.0$$

$$\text{Pool Elevation}[t-1] > Max \text{ Elevation for Power}[shutoff]$$

$$\text{Tailwater Elevation}[t-1] > \text{Max TailwaterElevation for Power}[shutoff]$$

$$Outflow[t-1] > Max \text{ Outflow for Power}[shutoff]$$

$$\text{Pool Elevation}[t-1] < \text{Minimum Power Elevation}$$

Note, the last equation is the behavior for minimum power pool. This slot is added by some of the power methods, not this method.
Also note, if Power Plant Cap Fraction is not zero but one of the other constraints is true, the power slots are set to zero, but the plant has not failed, so future timesteps can generate power.

If none of the above are true, the power method then proceeds as before using the computed **Power Plant Cap Fraction**.

17.1.6 Energy in Storage

The methods available in the Energy in Storage category are used to calculate the total energy that could be produced by the water stored in the reservoir.

17.1.6.1 None

Chosen if the user does not want to calculate the Energy In Storage. No slots are specifically associated with this method. This method performs no calculations.

17.1.6.2 EIS Table Lookup

The EIS TableLookup method obtains the amount of Energy In Storage from a table of Pool Elevation vs. Energy In Storage values and the Pool Elevation.

SLOTS SPECIFIC TO THIS METHOD

☛ **ENERGY IN STORAGE**

Type: AggSeriesSlot
Units: ENERGY
Description: Energy In Storage in the Reservoir
Information:
I/O: Output only
Links: Usually not linked

☛ **ENERGY IN STORAGE TABLE**

Type: TableSlot
Units: LENGTH VS. ENERGY
Description: Pool Elevation vs. Energy In Storage In the Reservoir
Information:
I/O: Required input
Links: Not linkable

The calculations involved with this method are very simple. The Pool Elevation for the current timestep is used to determine the Energy In Storage using the Energy In Storage Table. Simple linear interpolation is used.

17.1.6.3 EIS Table Lookup with Cons Pool

This method will be available only when the **Conservation Pool** or **Conservation and Flood Pools** method in the **Operating Levels** category is selected.

SLOTS SPECIFIC TO THIS METHOD

☛ **ENERGY IN STORAGE**

Type: AggSeriesSlot
Units: ENERGY
Description: Energy In Storage in the Reservoir

Information:
I/O: Output only
Links: Usually not linked

☛ ENERGY IN STORAGE TABLE

Type: TableSlot
Units: LENGTH VS. ENERGY
Description: Pool Elevation vs. Energy In Storage In the Reservoir
Information:
I/O: Required input
Links: Not linkable

☛ CONSERVATION POOL FULL EIS

Type: Series Slot
Units: MWH
Description: The EIS at the Top of the Conservation Pool
Information:
I/O: Output Only
Links: Not Linkable

The method is executed at the end of each dispatch method.

The **Pool Elevation** for the timestep is looked up in **Energy In Storage Table**. Simple linear interpolation is used. The resulting **Energy in Storage** is then set on the slot.

Next, the **Conservation Pool Full EIS** is calculated as follows:

$$\text{Top Conservation Pool Elevation}[t] = \text{Operating Level Table}[t, \text{Top of Conservation Pool Level}]$$

$$\text{Conservation Pool Full EIS}[t] = \text{Energy in Storage Table}[\text{Top of Conservation Pool Elevation}[t]]$$

Note, in the equation, the appropriate Operating Level Table will be used based on the timestep and the computation in the selected method in the **Conditional Operating Levels** category.

17.1.7 Tailwater

The Tailwater methods (available in the Tailwater category of the User Selectable Methods) calculate the Tailwater Elevation of a Level Power Reservoir. The Tailwater Elevation represents the water surface elevation immediately downstream of the Power Plant. This parameter is required to calculate the Operating Head of the Power Reservoir which is used to calculate either the Energy or Turbine Release of that Reservoir. The Tailwater methods are dependent upon the Reservoir being a Level Power Reservoir and a valid Power method (available in the Power category of the User Selectable Methods) being selected. These methods require a valid Outflow to perform their calculations.

17.1.7.1 None

This is the default method in the Tailwater category. This method performs no calculations. There are no slots specifically associated with this method.

Slots Specific to this Method

- **None**

17.1.7.2 Linked or Input

The Linked or Input method allows the user to either input values for the Tailwater Elevation or link the Tailwater Base Value to a slot (Pool Elevation or Backwater Elevation) on a downstream Reservoir. If the Tailwater Base Value is linked, the Tailwater Elevation cannot be input by the user. If the Tailwater Base Value is not linked, the Tailwater Elevation must be input by the user. The Tailwater Elevation is determined by following procedure if it is not input (i.e. the Tailwater Base Value is linked).

Slots Specific to this Method

- **Tailwater Base Value**

Type: Series
Units: LENGTH
Description: elevation of tailwater or base elevation used to compute elevation of tailwater
Information:
I/O: Optional; may be input or set by a link.
Links: May be linked to the Pool Elevation or Backwater Elevation of a downstream Reservoir.

The Linked or Input user method does not perform any calculations per se. There are however many logical evaluations performed in this method. The Linked or Input method is discussed below.

If a value for the Tailwater Base Value has been calculated by another user method or propagated through a link, the temporary value for the Tailwater Base Value (TWBaseValueTemp) is set equal to the calculated or linked Tailwater Base Value.

If the Tailwater Base Value is linked to another slot, the following steps are performed:

1. If the Tailwater Elevation is input, a **RiverWare**™ error is flagged and the run is aborted.

2. If TWBaseValueTemp is known, the Tailwater Elevation is set equal to the TWBaseValueTemp.

3. If TWBaseValueTemp is not known, but the previous timestep's Tailwater Elevation is known, the Tailwater Elevation is set equal to the previous timestep's Tailwater Elevation.

4. If neither the TWBaseValueTemp nor the previous timestep's Tailwater Elevation is known and the previous timestep was the initial timestep of the simulation, a **RiverWare**™ error is flagged and the simulation run is aborted.

5. If neither the TWBaseValueTemp nor the previous timestep's Tailwater Elevation is known and the previous timestep was not the initial timestep, the method is exited and waits for more information.

If the Tailwater Base Value is not linked to another slot, Tailwater Elevation must be input. If the Tailwater Elevation is not input, a **RiverWare**™ error is flagged and the simulation run is aborted.

17.1.7.3 Base Value Only

The Base Value Only method is similar to the Linked or Input method. If this method is selected, the user must either input values directly into the Tailwater Base Value slot or link the Tailwater Base Value slot to either the Pool Elevation slot or the Backwater Elevation slot of the downstream Reservoir. Either the initial Tailwater Elevation or the initial Tailwater Base Value must be input by the user for this method to execute successfully. The Tailwater Elevation for any timestep is computed as the average Tailwater Base Value over the timestep.

SLOTS SPECIFIC TO THIS METHOD

☛ **Tailwater Base Value**

Type: Series
Units: LENGTH
Description: elevation of tailwater or base elevation of tailwater is used to compute the tailwater elevation
Information:
I/O: Optional; may be input or linked.
Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

This method is based on logic similar to that of the Linked or Input method. The Base Value Only user method performs many logical evaluations to set the Tailwater Elevation. This method is described below.

The temporary Tailwater Base Value (TWBaseValueTemp) is set equal to the Tailwater Base Value if the Tailwater Base Value was calculated by another user method or set via a link.

If the Tailwater Base Value is linked to another slot, the following steps are performed:

1. If the previous timestep's Tailwater Base Value is not known, and both the previous timestep's Tailwater Elevation and TWBaseValueTemp are known, the Tailwater Elevation is calculated using the following formula:

$$\text{Tailwater Elevation} = \frac{(\text{TWBaseValueTemp} + \text{Tailwater Elevation}(-1))}{2}$$

where: Tailwater Elevation (-1) is the previous timestep's Tailwater Elevation.

2. If neither the previous timestep's Tailwater Base Value nor the TWBaseValueTemp is known but the previous timestep's Tailwater Elevation is known, the current timestep is set equal to the previous timestep's Tailwater Elevation.
3. If neither the previous timestep's Tailwater Base Value nor the previous timestep's Tailwater Elevation is known and the previous timestep is the initial timestep, a **RiverWare**™ error is posted and the simulation run is aborted.
4. If neither the previous timestep's Tailwater Base Value nor the previous timestep's Tailwater Elevation is known and the previous timestep is not the initial timestep, more information must be known. The method is exited and waits for more information to be known.
5. If both the previous timestep's Tailwater Base Value and the TWBaseValueTemp are known, the Tailwater Elevation is calculated using the following equation:

$$\text{Tailwater Elevation} = \frac{(\text{TWBaseValueTemp} + \text{Tailwater Base Value}(-1))}{2}$$

6. If the previous timestep's Tailwater Base Value is known but TWBaseValueTemp is not known, Tailwater Elevation is set equal to the previous timestep's Tailwater Base Value.

If the Tailwater Base Value is not linked, the following steps are performed:

1. If the Tailwater Elevation is not known and the TWBaseValueTemp is known, the Tailwater Elevation is set equal to the TWBaseValueTemp.
2. If neither the Tailwater Elevation nor the TWBaseValueTemp are known, a **RiverWare**™ error is flagged and the simulation run is aborted.

17.1.7.4 Base Value Plus Lookup Table

The Base Value Plus Lookup Table method computes the Tailwater Elevation. This is done by adding the average Tailwater Base Value (over the timestep) to a function of Outflow determined by the Tailwater Table. The Outflow value used to find the corresponding Tailwater value on the Tailwater Table is either the value of the local variable Flow or the value of the Outflow slot. The Tailwater Base Value may be input by the user or linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir. If the Tailwater Base Value is neither input nor linked, it is assumed to be zero.

Slots Specific to this Method

☛ **Tailwater Base Value**

Type: Series
Units: LENGTH
Description: the base elevation of the tailwater
Information:
I/O: Optional; can be input or linked.
Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

☛ **Tailwater Table**

Type: Table
Units: FLOW vs. LENGTH
Description: reservoir outflow vs. either the tailwater elevation or the tailwater elevation increment
Information: If the Tailwater Base Value is non-zero, the Tailwater Table gives values of incremental increase in Tailwater Elevation over the Base value. Otherwise, the table gives the Tailwater Elevation values.
I/O: Required input
Links: Not linkable

This method is based on logic similar to that of the Linked or Input method. The Base Value Plus Lookup Table user method performs many logical evaluations to set the Tailwater Elevation. This method is described below.

The temporary Tailwater Base Value (TWBaseValueTemp) is set equal to the Tailwater Base Value if the Tailwater Base Value was calculated by another user method or set via a link.

If the Local Info variable Flow is known, the local variable, tempflow, is set equal to Flow. If the Local Info variable Flow is not known, but the Outflow is known, tempflow is set equal to Outflow.

If the tempflow value is known, the following steps are taken to determine the Tailwater Elevation.

If the Tailwater Base Value is linked (and tempflow is known), the following steps are performed:

1. If both TWBaseValueTemp and the previous timestep's Tailwater Base Value are known, TWTemp (a local variable) is obtained from a table interpolation performed on the Tailwater Table using tempflow. The Tailwater Elevation is then calculated using the following equation.

$$\text{Tailwater Elevation} = (\text{TWBaseValueTemp} + \text{Tailwater Base Value }(-1))/2 + \text{TWTemp}$$

2. If TWBaseValueTemp is known but the previous timestep's Tailwater Base Value is known, but the previous timestep's Tailwater Basic Value is not known. TWTemp is obtained from a table interpolation performed on the Tailwater Table using tempflow. The Tailwater Elevation is then calculated using the following equation:

$$\text{Tailwater Elevation} = \text{TWBaseValueTemp} + \text{TWTemp}$$

3. If TWBaseValueTemp is not known but the previous timestep's Tailwater Base Value is known, TWTemp is obtained from a table interpolation performed on the Tailwater Table using tempflow. The Tailwater Elevation is then calculated using the following equation:

$$\text{Tailwater Elevation} = \text{Tailwater Base Value}(-1) + \text{TWTemp}$$

4. If neither TWBaseValueTemp nor the previous timestep's Tailwater Base Value are known but the previous timestep's Tailwater Elevation is known, the current timestep's Tailwater Elevation is set equal to the previous timestep's Tailwater Elevation.

5. If neither TWBaseValueTemp nor the previous timestep's Tailwater Base Value is known and the previous timestep is the initial timestep of the run, a **RiverWare**™ error is posted and the run is aborted.

6. If neither the TWBaseValueTemp nor the previous timestep's Tailwater Base Value is known and the previous timestep is not the initial timestep of the run, the method is exited and waits for more information.

If the Tailwater Base Value is NOT linked (and tempflow is known), the following steps are performed:

1. If TWBaseValueTemp is known, TWTemp is obtained from a table interpolation performed on the Tailwater Table using tempflow. The Tailwater Elevation is then calculated using the following equation:

$$\text{Tailwater Elevation} = \text{TWBaseValueTemp} + \text{TWTemp}$$

2. If TWBaseValueTemp is not known, the Tailwater Elevation is obtained from a table interpolation performed on the Tailwater Table using tempflow.

The only case where tempflow is not known is when Outflow is not known. This scenario only occurs at the first timestep in one of the three “given Energy” dispatch methods. The following steps are performed to determine the Tailwater Elevation if tempflow is not known.

If the Tailwater Base Value is linked (and tempflow is not known), the following steps are taken:

1. If TWBaseValueTemp and the previous timestep’s Tailwater Elevation are known, the Tailwater Elevation is calculated using the following equation:

$$\text{Tailwater Elevation} = (\text{TWBaseValueTemp} + \text{Tailwater Elevation}(-1))/2$$

2. If both TWBaseValueTemp and the previous timestep’s Tailwater Base Value are known but the previous timestep’s Tailwater Elevation is not known, Tailwater Elevation is calculated using the following equation:

$$\text{Tailwater Elevation} = (\text{TWBaseValue Temp} + \text{Tailwater Base Value}(-1))/2$$

3. If TWBaseValueTemp is known, but neither the previous timestep’s Tailwater Elevation nor the previous timestep’s Tailwater Base Value are known, Tailwater Elevation is set equal to TWBaseValueTemp.

4. If TWBaseValueTemp is not known but the previous timestep’s Tailwater Elevation is known, the current timestep’s Tailwater Elevation is set equal to the previous timestep’s Tailwater Elevation.

5. If neither TWBaseValueTemp nor the previous timestep’s Tailwater Elevation are known, but the previous timestep’s Tailwater Base Value is known, Tailwater Elevation is set equal to the previous timestep’s Tailwater BaseValue.

6. If the previous timestep is the initial timestep of the run, and none of the following are known: TWBaseValueTemp, the previous timestep’s Tailwater Base Value, and the previous timestep’s Tailwater Elevation, a **RiverWare**™ error is posted and the simulation run is aborted.

7. If the previous timestep is NOT the initial timestep of the run, and none of the following are known: TWBaseValueTemp, the previous timestep’s Tailwater Base Value, and the previous timestep’s Tailwater Elevation, the method is exited and waits for more information to execute.

If the Tailwater Base Value is not linked (and tempflow is NOT known), the following procedures are performed:

1. If the Tailwater Elevation is known, the method is exited because no calculations need to be performed.
2. If the Tailwater Elevation is NOT known, and TWBaseValueTemp is NOT equal to zero, Tailwater Elevation is set equal to TWBaseValueTemp.
3. If the Tailwater Elevation is NOT known and either the temporary Tailwater Base Value is NOT known or equal to zero or both, Tailwater Elevation is set equal to the elevation corresponding to zero on the Tailwater Table.

Definitions of some of the terms used in the above equations are located below:

- TWBaseValueTemp: a temporary value for the Tailwater Base Value. This value may be determined from another method or propagated across a link.
- tempflow: a local variable used as the outflow from the Reservoir.
- TWTemp: a local variable used to represent the incremental increase in the Tailwater Elevation over the Tailwater Base Value.
- Tailwater Elevation (-1): the Tailwater Elevation at the previous timestep.
- Tailwater Base Value (-1): the Tailwater Base Value at the previous timestep.

17.1.7.5 Stage Flow Lookup Table

The Stage Flow Lookup Table method is similar to the other methods for determining Tailwater Elevation. The Tailwater Elevation is obtained from a 3-dimensional table relating Outflow, Downstream Stage, and the corresponding Tailwater Elevation for most cases. The data in this table must be input by the user. The value for Downstream Stage that is used in this method is the larger of the Tailwater Reference Elevation or the Tailwater Base Value if the Tailwater Base Value is linked to a downstream elevation. If the Tailwater Base Value is not linked, the temporary Tailwater Base Value is used as the Downstream Stage. An average value for the Tailwater Elevation over the timestep is used whenever possible.

Slots Specific to this Method

☛ **Stage Flow Tailwater Table**
Type: Table
Units: FLOW vs. LENGTH vs. LENGTH
Description: Reservoir Outflow vs. Downstream Elevation (Tailwater Base Value) vs. Tailwater Elevation
Information: Data must be entered into the table in increasing blocks of the same Outflow value for the 3-dimensional table interpolator to work correctly. For every

block of same Outflows in column 1, Stages should be listed in increasing order in column 2, and the corresponding Tailwater Elevations in column 3.

Outflow	Downstream Stage	TW Elevation
100	500	510
100	550	560
100	600	610
200	500	520
200	550	570
200	600	620
300	500	530
300	550	580
300	600	630

I/O: Required input
Links: Not linkable

☛ Tailwater Base Value

Type: Series
Units: LENGTH
Description: base elevation of the tailwater
Information:
I/O: Optional; can be input or set by a link.
Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

☛ Tailwater Reference Elevation

Type: Table
Units: LENGTH
Description: lowest Reservoir discharge Elevation when there are no backwater effects from a downstream pool (reservoir)
Information: If this slot has input data, the greater of the Tailwater Reference Elevation or the linked lower reservoir's Pool or Backwater Elevation is used to calculate the Tailwater Base Value. If the Tailwater Base Value is linked to a downstream elevation, this value must be specified by the user. If the Tailwater Base Value is not linked, this value is not used, even if specified.
I/O: Required input
Links: Not linkable

The Stage Flow Lookup Table user method performs many logical tests to determine the Tailwater Elevation. This method is described below.

The first step in this method is to check and see if the Tailwater Reference Elevation is known. If the Tailwater Reference Elevation is not known, a **RiverWare**™ error is posted and the run is aborted.

The temporary Tailwater Base Value (TWBaseValueTemp) is set equal to the Tailwater Base Value if the Tailwater Base Value was calculated by another user method or set via a link.

If the local variable Flow is known, the local variable tempflow is set equal to Flow If the local variable Flow is not known, but the Outflow is known, if the Local Info variable Flow is not known but Outflow is known, tempflow is set equal to Outflow.

If the tempflow value is known, the following steps are taken to determine the Tailwater Elevation.

If the Tailwater Base Value is linked (and tempflow is known), the following steps are performed:

1. If both TWBaseValueTemp and the previous timestep's Tailwater Base Value are known, downstreamStage is computed as the average of the previous timestep's Tailwater Base Value and either the Tail Water Reference Elevation or TWBaseValueTemp (whichever is greater). A table interpolation is performed to determine the Tailwater Elevation using the downstreamStage, tempflow, and the Stage Flow Tailwater Table.

2. If TWBaseValueTemp is known but the previous timestep's Tailwater Base Value is NOT known, downstreamStage is set equal to either the Tail Water Reference Elevation or TWBaseValueTemp (whichever is greater). A table interpolation is performed to determine the Tailwater Elevation using the downstreamStage, tempflow, and the Stage Flow Tailwater Table.

3. If the previous timestep's Tailwater Base Value is known but TWBaseValueTemp is NOT known, downstreamStage is set equal to the previous timestep's Tailwater Base Value. A table interpolation is performed to determine the Tailwater Elevation using the downstreamStage, tempflow, and the Stage Flow Tailwater Table.

4. If neither TWBaseValueTemp nor the previous timestep's Tailwater Base Value are known, but the previous timestep's Tailwater Elevation is known, the current timestep's Tailwater Elevation is set equal to the previous timestep's Tailwater Elevation.

5. If the previous timestep was the initial timestep and none of the following are known: TWBaseValueTemp, the previous timestep's Tailwater Base Value, and the previous timestep's Tailwater Elevation, a **RiverWare**™ error is posted and the run is aborted.

6. If the previous timestep was NOT the initial timestep, and none of the following are known: the temporary TWBaseValueTemp, the previous timestep's Tailwater Base Value, and the previous timestep's Tailwater Elevation, the method is exited and waits for more information.

If the Tailwater Base Value is not linked (and tempflow is known), the following steps are performed:

1. If TWBaseValueTemp is known, downstreamStage is set equal to TWBaseValueTemp. A table interpolation is performed to determine the Tailwater Elevation using the downstreamStage, tempflow, and the Stage Flow Tailwater Table.

2. If TWBaseValueTemp is NOT known, a **RiverWare**™ error is posted and the run is aborted.

The only case where tempflow is not known is when Outflow is not known. This scenario only occurs at the first timestep in one of the three “given Energy” dispatch methods. The following steps are performed to determine the Tailwater Elevation if tempflow is not known.

If the Tailwater Base Value is linked (and tempflow is not known), the following steps are performed:

1. If both TWBaseValueTemp and the previous timestep’s Tailwater Elevation are known, the Tailwater Elevation is calculated using the following equation:

$$\text{Tailwater Elevation} = (\text{TWBaseValue Temp} + \text{Tailwater Elevation }(-1))/2$$

2. If both TWBaseValueTemp and the previous timestep’s Tailwater Base Value are known but the previous timestep’s Tailwater Elevation is not known, Tailwater Elevation is computed using the following equation:

$$\text{Tailwater Elevation} = (\text{TWBaseValue Temp} + \text{Tailwater Base Value }(-1))/2$$

3. If TWBaseValueTemp is known but neither the previous timestep’s Tailwater Elevation nor the previous timestep’s Tailwater Base Value are known, Tailwater Elevation is set equal to TWBaseValueTemp.

4. If TWBaseValueTemp is not known but the previous timestep’s Tailwater Elevation is known, the current timestep’s Tailwater Elevation is set equal to the previous timestep’s Tailwater Elevation.

5. If neither TWBaseValueTemp nor the previous timestep’s Tailwater Elevation is known, but the previous timestep’s Tailwater Base Value is known, Tailwater Elevation is set equal to the previous timestep’s Tailwater Base Value.

6. If the previous timestep is the initial timestep of the run, and none of the following are known: TWBaseValueTemp, the previous timestep’s Tailwater Base Value, and the previous timestep’s Tailwater Elevation, a **RiverWare**™ error is posted, and the run is aborted.

7. If the previous timestep is NOT the initial timestep of the run and none of the following are known: TWBaseValueTemp, the previous timestep's Tailwater Base Value, and the previous timestep's Tailwater Elevation, the method is exited and waits for more information.

If the Tailwater Base Value is not linked (and tempflow is not known) the following procedures are performed:

1. If the Tailwater Elevation is known, the method is exited because no calculations need to be performed.

2. If the Tailwater Elevation is NOT known and the TWBaseValueTemp is NOT equal to zero, Tailwater Elevation is set equal to TWBaseValueTemp.

3. If the Tailwater Elevation is NOT known and either the temporary Tailwater Base Value is NOT known or equal to zero or both, Tailwater Elevation is set equal to the elevation corresponding to zero on the Tailwater Table.

Definitions of some of the terms used in the above equations are located below:

- ***TWBaseValueTemp* -** a temporary value for the Tailwater Base Value. This value may be determined from another method or propagated across a link.
- ***tempflow* -** a local variable used as the outflow from the Reservoir.
- ***downstreamStage* -** a local variable used to hold the value of the downstream stage elevation.
- ***Tailwater Elevation (-1)* -** the Tailwater Elevation at the previous timestep.
- ***Tailwater Base Value (-1)* -** the Tailwater Base Value at the previous timestep.

17.1.7.6 Compare to Avg Base Value

The Compare to Avg Base Value method uses similar methodology as the Base Value Plus Lookup Table and Stage Flow Lookup Table methods. The only difference between this method and the others is that a new local variable, TWCompare, is used. The value for TWCompare is compared with a table lookup value for the Tailwater Elevation. The larger of the two values is used as the Tailwater Elevation. An average Tailwater Elevation over the timestep is calculated whenever possible.

Slots Specific to this Method

☛ **Tailwater Base Value**
Type: Series
Units: LENGTH
Description: base elevation of the tailwater
Information:
I/O: Optional; can be input of set by a link.

Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

☛ **Tailwater Table**
Type: Table
Units: FLOW VS. LENGTH
Description: Reservoir Outflow vs. either the Tailwater Elevation or the tailwater elevation increment
Information: If the Tailwater Base Value is non-zero, the Tailwater Table holds values of incremental increase in Tailwater Elevation over the Base value. Otherwise, the table holds the Tailwater Elevation values.
I/O: Required input
Links: Not linkable

The Compare to Avg Base Value user method performs many logical evaluations to determine the Tailwater Elevation. This method is described below.

The temporary Tailwater Base Value (TWBaseValueTemp) is set equal to the Tailwater Base Value is the Tailwater was calculated by another user method or set via a link. For the Tailwater Compare Method, the Tailwater Base Value must be linked to another object. Otherwise there would be no reason to compare values. If the Tailwater Base Value is not linked, a **RiverWare**™ error is posted and the simulation run is aborted.

If the Local Info variable Flow is known, the local variable tempflow is set equal to Flow.

If the local variable Flow is not known, but Outflow is known, tempflow is set equal to Outflow.

If the tempflow value is known, the following steps are taken to determine the Tailwater Elevation.

1. If both TWBaseValueTemp and the previous timestep's Tailwater Base Value are known,

$$TWCompare = (TWBaseValueTemp + Tailwater\ Base\ Value\ (-1))/2$$

2. If TWBaseValueTemp is known but the previous timestep's Tailwater Base Value is NOT known, TWCompare is set equal to TWBaseValueTemp.

3. If TWBaseValueTemp is not known but the previous timestep's Tailwater Base Value is known, TWCompare is set equal to the previous timestep's Tailwater Base Value.

4. If neither TWBaseValueTemp nor the previous timestep's Tailwater Base Value are known and the previous timestep is the initial timestep, a **RiverWare**™ error is posted and the run is aborted.

5. If neither TWBaseValueTemp nor the previous timestep's Tailwater Base Value are known and the previous timestep is NOT the initial timestep, the method is exited and waits for more information.

Once the TWCompare value has been determined, a table interpolation is performed using the Tailwater Table to obtain the tailwater elevation corresponding to the tempflow value. The tailwater elevation value obtained from this interpolation is then compared to TWCompare. The largest of the two values is used to set the Tailwater Elevation.

The only case where tempflow is not known is when Outflow is not known. This scenario only occurs at the first timestep in one of the three "given Energy" dispatch methods. The following steps are performed to determine the Tailwater Elevation if tempflow is not known.

1. If both TWBaseValueTemp and the previous timestep's Tailwater Elevation are known,

$$\text{Tailwater Elevation} = (\text{TWBaseValue Temp} + \text{Tailwater Elevation (-1)})/2$$

2. If both TWBaseValueTemp and the previous timestep's Tailwater Base Value are known but the previous timestep's Tailwater Elevation is not known,

$$\text{Tailwater Elevation} = (\text{TWBaseValue Temp} + \text{Tailwater Base Value (-1)})/2$$

3. If TWBaseValueTemp is linked and the temporary Tailwater Base Value is known but neither the pervious timestep's Tailwater Elevation nor the previous timestep's Tailwater Base Value are known, the Tailwater Elevation is set equal to TWBaseValueTemp.

4. If TWBaseValueTemp is not known but the previous timestep's Tailwater Elevation is known, the current timestep's Tailwater Elevation is set equal to the previous timestep's Tailwater Elevation.

5. If neither TWBaseValueTemp nor the previous timestep's Tailwater Elevation are known, but the previous timestep's Tailwater Base Value is known, the current timestep's Tailwater Elevation is set equal to the previous timestep's Tailwater Base Value.

6. If the previous timestep is the initial timestep of the run, and none of the following are known: TWBaseValueTemp, the previous timestep's Tailwater Base Value, and the previous timestep's Tailwater Elevation, a **RiverWare**™ error is posted and the simulation is aborted.

7. If the previous timestep is NOT the initial timestep of the run and none of the following are known: TWBaseValueTemp, the previous timestep's Tailwater Base Value, and the previous timestep's Tailwater Elevation, the method is exited and waits for more information.

Definitions of some of the terms used in the above equations are located below:

- **TWBaseValueTemp**: a temporary value for the Tailwater Base Value. This value may be determined from another method or propagated across a link.
- **tempflow**: a local variable used as the outflow from the Reservoir.
- **TWCompare**: a local variable used to hold a value that is compared to the Tailwater Elevation obtained by table interpolation.
- **Tailwater Elevation (-1)**: the Tailwater Elevation at the previous timestep.
- **Tailwater Base Value (-1)**: the Tailwater Base Value at the previous timestep.

17.1.7.7 Coefficients Table

The Coefficients Table method multiplies Outflow, Tailwater Base Value, and Tailwater Elevation at the current and/or previous timestep by coefficients that are stored on a table. These products are added together along with constants to compute the Tailwater Elevation.

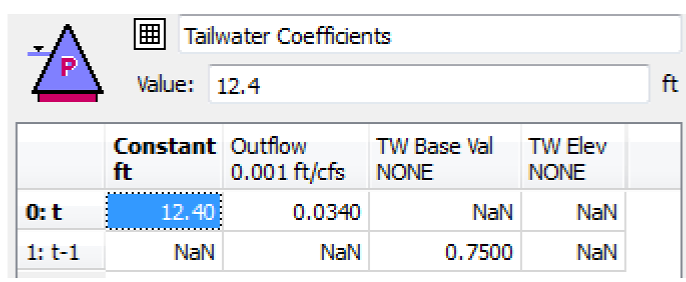
Slots Specific to this Method

☛ Tailwater Base Value

Type: Series
Units: LENGTH
Description: base elevation of the tailwater
Information:
I/O: Optional; can be input or set by a link.
Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

☛ Tailwater Coefficients

Type: Table
Units: LENGTH, LENGTH/FLOW, NONE, NONE
Description: This table contains the coefficients used in the calculation. The columns are as follows: **Constant**, **Outflow**, **TW Base Val**, **TW Elev**. The first row (**t**) represents the value to multiply by the current timestep's value. The second row (**t-1**)represents the value to multiply by the previous timestep's value.



Information: Not every cell will have a value, but there must be at least one value in the table. In addition, the coefficient for **TW Elev** at t cannot be non-zero or an error will be issued. (Tailwater Elevation at the current timestep is the value being computed, so it is not possible to use it in the calculation). If any coefficents are specified (non-zero) for t-1 for **Outflow**, **TW Base Val**, or **TW Elev**, then the initial timestep value of those slots must also be specified.
I/O: Input only
Links: Not Linkable

At the beginning of the run, the **Tailwater Coefficients** table is verified for valid coefficients as described in the slot information above. Not every coefficient has to be specified.

When the tailwater method is executed, the following computation is performed.

$$\begin{aligned}\text{Tailwater Elevation} = {} & \text{Constant Coeff}[t] + \text{Constant Coeff}[t-1] + \\ & \text{Outflow Coeff}[t] \times \text{flow} + \text{Outflow Coeff}[t-1] \times \text{Outflow}[t-1] + \\ & \text{TW Base Val Coeff}[t] \times \text{TailwaterBaseValueTemp}[t] + \\ & \text{TW Base Val Coeff}[t-1] \times \text{Tailwater BaseValue}[t-1] + \\ & \text{TW Elev Coeff}[t-1] \times \text{Tailwater Elevation}[t-1]\end{aligned}$$

Each coefficient and each series value are checked for validity before including in the total.

In the above equation, the values for **flow** and **TailwaterBaseValueTemp** are based on the information that is known:

- The *flow* variable represents the outflow passed into the method if valid, or the value on the Outflow slot if not passed in. If neither is valid, Tailwater Elevation is set to the previous Tailwater Elevation. If that is not valid, the method exits and waits for a flow value to become valid.

The *TailwaterBaseValueTemp* is either the base value passed into the method or the value on the Tailwater Base Value slot. If neither is known, the previous Tailwater Base Value is used. If that is not valid, the method executes to wait for more information.

17.1.7.8 Hoover Tailwater

The Hoover Tailwater method was developed by the USBR for use on Lake Mead. This method uses a table of empirically derived constants.

Slots Specific to this Method

☛ **Tailwater Base Value**
Type: Series
Units: LENGTH
Description: base elevation of the tailwater
Information:

I/O: Optional; can be input or set by a link.
Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

Tailwater Table

Type: Table
Units: FLOW VS. LENGTH
Description: Reservoir Outflow vs. either the Tailwater Elevation or the tailwater elevation increment
Information: If the Tailwater Base Value is non-zero, the Tailwater Table holds values of incremental increase in Tailwater Elevation over the Base value. Otherwise, the table holds the Tailwater Elevation values.
I/O: Required input
Links: Not linkable

Hoover Tailwater Table

Type: Table
Units: NO UNITS
Description: constants for equation that calculates the Tailwater Elevation
Information: This method was developed by the USBR for use on Lake Mead. Column 1 is the TWaverage in feet, column 2 is hte5, column 3 is coeff1.
I/O: Required input
Links: Not linkable

The Hoover Tailwater user method performs calculations to determine the Tailwater Elevation. This method is described below:

The first step of this method is to set the temporary Tailwater Base Value (TWBaseValueTemp). TWBaseValueTemp is set equal to the Tailwater Base Value if the Tailwater Base Value was calculated by another user method or set via a link.

Then the previous timestep's Tailwater Base Value is checked. If the previous timestep's Tailwater Base Value is not known, a **RiverWare**™ error is posted and the run is aborted.

The TWaverage, a local variable, is then calculated using the following steps:

1. If TWBaseValueTemp is known,

$$\text{TWaverage} = (\text{TWBaseValueTemp} + \text{Tailwater Base Value }(-1))/2$$

2. If the temporary TWBaseValueTemp is NOT known, the TWaverage is set equal to the previous timestep's Tailwater Base Value.

Once the TWaverage has been calculated, it is used to interpolate the empirical constants for the following equation from the Hoover Tailwater Table.

Finally, $\text{TELEFT} = \text{hte5} + ((\text{Qcfs})/1000 - 5) \times \text{coeff1}$

Tailwater Elevation is then set equal to TELEFT (which is converted into internal units).

Definitions of some of the terms used in the above equations are located below:

- **TWBaseValueTemp**: a temporary value for the Tailwater Base Value. This value may be determined from another method or propagated across a link.
- **Tailwater Base Value (-1)**: the Tailwater Base Value at the previous timestep.
- **TWaverage**: a local variable that is used as the average of the Tailwater Elevation over the timestep.
- **TELEFT**: the Tailwater Elevation in feet as determined by the equation given above.
- **hte5**: an empirical constant developed by the USBR.
- **Qcfs**: the flow in cfs.
- **coeff1**: an empirical constant developed by the USBR.

17.1.8 Spill

The Spill methods (except the Monthly Spill which is described in Section 17.1.8.2) calculate the Spill from Reservoirs based on several possible physical combinations of controlled and uncontrolled spillways.

The **Regulated Spill** and **Bypass** slots are regulated (i.e. controlled) spill structures. Values in these two slots can be specified by the user via inputs or rules. Each slot accommodates spill up to the maximum amount as specified by its rating table (**Regulated Spill Table** and **Bypass Table**). **Unregulated Spill** is an uncontrolled spill. Therefore, it is always a computed output based on the average **Pool Elevation** of the reservoir as specified in the **Unregulated Spill Table**. Thus, the user selects a **Spill** method based on the combination of structures (Regulated Spill, Bypass, and/or Unregulated Spill) that exist on the reservoir and the level of granularity desired.

The total **Spill** slot is the sum of the individual spills from each structure. Spills are calculated twice for each timestep. The first time a **Spill** method is called from a dispatch method, it checks for user inputs, calculates any **Unregulated Spill**, and sets the spill to zero for **Regulated Spill** and **Bypass** structures where there is no user-specified value. The total **Spill** is then calculated and returned to the dispatch method. The dispatch method determines (**Turbine**) **Release** by subtracting the **Spill** from **Outflow**, and executes the user-specified power calculation method. (On the power reservoirs, the slot is called **Turbine Release**, on the storage reservoir, the slot is called **Release**. In this description, we use the term (**Turbine**) **Release**.) If the (**Turbine**) **Release** cannot be met in the power calculation method, a second call is made to the spill calculation method. The excess flow is then distributed among the **Regulated Spill** and/or **Bypass** structures which have available capacity. If both **Regulated Spill** and **Bypass** are available, excess spill is typically first discharged through the **Regulated Spill** (except when the “**Bypass, Regulated and Unregulated**” method is selected; for this method the **Bypass** gets spill first).

The optional DRIFT flag is available on the **Regulated Spill** and **Bypass** slots. When the DRIFT flag is set for several sequential timesteps, the method models varying flow through a set spillway gate in response to fluctuations in **Pool Elevation**. The first timestep prior to initializing drift is used to determine a gate index called **Regulated** (or **Bypass**) **Drift Index**. This index is interpolated from the 3-dimensional **Regulated** (or **Bypass**) **Spill Index Table**, which relates **Pool Elevation** to **Spill** for several gate indices. In the subsequent timesteps where the DRIFT flag is set, the same index is used to find the spill value at the current average **Pool Elevation**. The gate index is maintained throughout the selected time period. At each timestep, a new value of spill is calculated for the structure based on the current **Pool Elevation**. Specifying DRIFT is considered an input, and may affect over determination of spill parameters.

17.1.8.1 None

None is used if **Spill** should not be modeled. In this method, the **Spill** slot on the reservoir is set equal to zero. All releases must be through the (**Turbine**) **Release**.

Slots Specific to this Method

☛ None

This method sets the **Spill** to zero and performs no further calculations. If the method is being called for the second time in a timestep because there is excess outflow that won't fit through (**Turbine**) **Release**, an error will be posted which states, "No Spillways Available." In this case, either decrease the **Outflow** or select a different spill method.

17.1.8.2 Monthly Spill

The Monthly Spill method is only appropriate for use in long timestep models where Reservoir fluctuations over the timestep cannot be accurately determined. It is important to note that there is no physical (head dependent) basis to the spill in this method. In this method, there are three components to the spill: unregulated, regulated and bypass. Both Regulated Spill and Bypass are considered controlled releases. A Maximum Controlled Release must be specified by the user.

$$Maximum\ Controlled\ Release = max\ Turbine\ Release + max\ Regulated\ Spill + Bypass$$

Any additional Outflow is immediately categorized as Unregulated Spill.

$$Unregulated\ Spill = Max(0.0, Outflow - Maximum\ Controlled\ Release)$$

Bypass may be specified as a user input. If not input, **RiverWare™** sets this slot to zero. Regulated Spill is always computed by **RiverWare™**. It is set to zero unless the Reservoir cannot release the Outflow through the Release and Bypass. When this occurs, the additional portion of the Outflow is released through the Regulated Spill.

Slots Specific to this Method

☛ Bypass

Type: Series Slot
Units: FLOW
Description: flow through the Bypass spillway
Information:
I/O: Optional; may be input by the user or set to zero by **RiverWare™**.
Links: Usually not linked

☛ Maximum Controlled Release

Type: Table Slot
Units: FLOW
Description: the maximum amount of Turbine Flow, Regulated Spill, and Bypass
Information: 1X1 table slot
I/O: Required input
Links: Not linkable

☛ Regulated Spill

Type: Series Slot
Units: FLOW
Description: excess Outflow not released through the turbine(s)
Information:
I/O: Output only
Links: Usually not linked

☛ Unregulated Spill

Type: Series Slot
Units: FLOW
Description: Outflow in excess of the Maximum Controlled Release
Information:
I/O: Output only
Links: Not linkable

Initially, the Monthly Spill method is called before Release has been calculated. If the Outflow is greater than the Maximum Controlled Release, Unregulated Spill is set as:

$$Unregulated\ Spill = Outflow - Maximum\ Controlled\ Release$$

If the Outflow is less than the Maximum Controlled Release, Unregulated Spill is set to zero. In both cases, all of the following evaluations are also made:

$$Regulated\ Spill = 0.0$$

$$Bypass = user\ input\ or\ 0.0$$

$$Spill = Unregulated\ Spill + Regulated\ Spill + Bypass$$

After Release is calculated, the Monthly Spill method may be called a second time. The method is called a second time if the Release cannot accommodate the remaining portion of the Outflow:

$$Outflow > Unregulated\ Spill + Release + Bypass$$

Remember that Unregulated Spill was calculated before the Release was calculated.

If this occurs, Regulated Spill and Spill are reevaluated as follows:

$$Regulated\ Spill = Outflow - Unregulated\ Spill - Release - Bypass$$

$$Spill = Unregulated\ Spill + Regulated\ Spill + Bypass$$

17.1.8.3 Unregulated

The Unregulated spill method models a single uncontrolled spillway called **Unregulated Spill**. The **Unregulated Spill** is a function of the average reservoir **Pool Elevation**. Because it is uncontrolled, it takes precedence (i.e. water goes through it first) over other types of outflow (i.e. **Release** or **Turbine Release**) in the reservoir. When this method is chosen the user category Unregulated Spill Type , Section 17.1.9, will appear.

The user may not specify (input or via rules) any spill slots with this method.

Slots Specific to this Method

☛ Unregulated Spill

Type: Series Slot
Units: FLOW
Description: spill corresponding to the average Pool Elevation over the timestep
Information:
I/O: Output only
Links: Not linkable

☛ Unregulated Spill Capacity Fraction

Type: Series Slot
Units: Decimal
Description: The fraction of the Unregulated Spill structure that is available.
Information: If not input or set by a rule, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 50 ft of a 1000 ft long crest is blocked, the Unregulated Spill Capacity Fraction would be input to 0.95.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Unregulated Spill Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation versus corresponding Unregulated Spill values

Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required input
Links: Not linkable

When the Unregulated spill method is called for the first time from the Dispatch Method, Unregulated Spill is calculated and Spill is set equal to Unregulated Spill.

THE STEPS FOR COMPUTING UNREGULATED SPILL ARE GIVEN BELOW:

1. A temporary variable called "initHW" is created to represent the Pool Elevation at the beginning of the timestep. Likewise, "endHW" is created to represent the Pool Elevation at the end of the timestep (If the Pool Elevation at the end of the timestep is not known, endHW is set equal to the Pool Elevation at the beginning of the timestep.)

2. The Unregulated Spillway Crest is set equal to the Pool Elevation that corresponds to a Spill of zero (from the Unregulated Spill Table).

3. If both initHW and endHW are less than or equal to the Unregulated Spillway Crest, Unregulated Spill is set equal to zero.

4. If both initHW and endHW are greater than or equal the Unregulated Spillway Crest, the average Pool Elevation is used to determine the Unregulated Spill from the Unregulated Spill Table.

$$\text{Unregulated Spill} = \text{Value from table} \times \text{Unregulated Spill Capacity Fraction}$$

5. If either initHW or endHW is greater than the Unregulated Spillway Crest and the other is lower than the crest, the following evaluations and computations are performed:

$$\text{maxHW} = \text{the greater of initHW and endHW}$$

$$\text{minHW} = \text{the lesser of initHW and endHW}$$

$$\text{avgHW} = \frac{\text{maxHW} + \text{Unregulated Spillway Crest}}{2}$$

$$\text{spill fraction} = \frac{\text{maxHW} - \text{Unregulated Spillway Crest}}{\text{maxHW} - \text{minHW}}$$

That is, spill fraction corresponds to the fraction of the timestep during which spill occurs.

A temporary variable called "temp spill" is obtained from the linear interpolation of the Unregulated Spill Table using avgHW. Unregulated Spill is then calculated as:

$$\text{Unregulated Spill} = \text{spill fraction} \times \text{temp spill} \times \text{Unregulated Spill Capacity Fraction}$$

The Unregulated Spill is then limited to be less than or equal to the maxUnregulatedSpill if one has been calculated. See below.

The Unregulated spill method will be called a second time (after the (**Turbine**) **Release** has been calculated) only if the sum of (**Turbine**) **Release** and **Spill** are less than **Outflow**. When this is the case, an error which reads, "Outflow greater than spillway capacities and Release" is posted because the excess **Spill** cannot be incorporated.

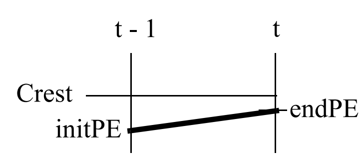
Computing the Maximum Possible Unregulated Spill

If an unregulated spill method is called from the solveMB_givenInflowRelease dispatch method, getMinSpillGivenInflowRelease or solveTurbineRelGivenEnergyInflow RPL predefined functions, there is an upper limit on the unregulated spill. The upper limit prevents the unregulated spill from dropping the reservoir below the crest. The following algorithm is performed at the start of the following methods to compute this maximum unregulated spill (maxUnregulatedSpill):

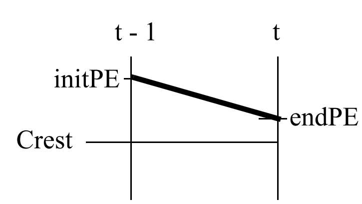
- solveMB_givenInflowRelease dispatch method (Storage and Level Power reservoirs)
- solveMB_givenInflowOutflow dispatch method (Storage and Level Power reservoirs)
- getMinSpillGivenInflowRelease RPL predefined function
- solveTurbineRelGivenEnergyInflow RPL predefined function

1. Given previous Storage and Pool Elevation (initPE), all known inflows (Inflow, Hydrologic Inflow, Precipitation, Return Flow, etc) and known outflows (Release, Regulated Spill, Bypass, Evaporation, Diversion, etc) are used to compute the storage and pool elevation (endPE) that would occur with no additional unregulated spill.

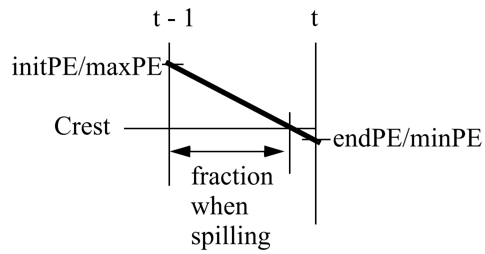
2. If both initPE and endPE are less than or equal to the Unregulated Spillway Crest, maxUnregulatedSpill is set to zero and the computation exits. There is no way there could be unregulated spill. See figure to the right.



3. If both initPE and endPE are greater than or equal to the Unregulated Spillway Crest, the maxUnregulatedSpill is computed as the flow that would draw the reservoir down to exactly reach the crest at the end of the timestep. This computation solves the reservoir mass balance and includes all source and sink terms. All water above the crest could be spilled. See figure to the right.



4. If either initPE or endPE is greater than the Unregulated Spillway Crest and the other is lower than the crest (because of existing Diversions, Evap, etc), the following evaluations and computations are performed: (See figure to the right)



$$\text{maxPE} = \text{the greater of initPE and endPE}$$

$$\text{minPE} = \text{the lesser of initPE and endPE}$$

$$\text{spill fraction} = \frac{\text{maxPE} - \text{Unregulated Spillway Crest}}{\text{maxPE} - \text{minPE}}$$

That is, spill fraction corresponds to the fraction of the timestep during which spill occurs.

5. The storage at the crest, crestStorage, is computed from the Elevation Volume Table.

6. The storage, maxStorage, that corresponds to the maxPE is found on the Elevation Volume Table.

7. The maxUnregulatedSpill (limited to be greater than or equal to zero) is then computed as:

$$\text{maxUnregulatedSpill} = \frac{\text{maxStorage} - \text{crestStorage}}{\text{TimestepLength}} \times \text{spillFraction}$$

The maxUnregulatedSpill is applied as a final limit on the Unregulated Spill. Remember, the Unregulated Spill is computed as described in the start of this method using the Unregulated Spill Table and a similar spill fraction approach; it may already be less than the maxUnregulatedSpill.

17.1.8.4 Regulated

The Regulated spill method models **Spill** using one controlled spillway called **Regulated Spill**. Because the spill is controlled, the spill may be any value between zero and the maximum possible regulated spill for that pool elevation. The user may specify (input or via rules) either:

- No slots
- **Spill** or
- **Regulated Spill**

If either is specified and there is excess flow which cannot be met by the (**Turbine**) **Release**, a **RiverWare**™ error will be flagged and the simulation halted.

Slots Specific to this Method

☛ Regulated Spill

Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare™**. If Regulated Spill is input by the user and the value is less than the required spill, a **RiverWare™** error is flagged and the simulation is halted.
Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☛ Regulated Spill Capacity Fraction

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Regulated Spill structure that is available.
Information: If not input or set by a rule, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Regulated Spill Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index
Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Regulated Spill Index Table.
Links: Not linkable

☛ Regulated Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding Maximum Regulated Spill values
Information:
I/O: Required input
Links: Not linkable

☛ Regulated Spill Index Table

Type: Table Slot

Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Regulated Spill
Information: Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column 3. The table shown below is an example of the proper way to formulate the Regulated Spill Index Table.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130
4	550	180
4	600	230

I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.
Links: Not linkable

Note: Regulated Spill and Spill are both outputs if neither is specified by the user. Only one of these slots, however, can be specified on a given timestep. When this is the case, the other slot will be output.

The first step in the Regulated spill method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 17.1.10.2. If not, the minRegSpill is zero. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

Release is then checked to see if it has been calculated. If Release is not known, it means that the method is being called for the first time for the particular timestep and the following steps are taken:

1. If both the Spill and Regulated Spill are input (remember setting a DRIFT flag is considered an input), a **RiverWare**™ error is flagged and the run is aborted.

2. If Spill is input by the user, and it is greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted. Otherwise, Regulated Spill is set equal to Spill.

3. If Regulated Spill is input/rules by the user and the DRIFT flag is set, a function is called to perform the drift calculations. A description of the DRIFT function is given at the end of this method description. If Regulated Spill is input and greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted. Otherwise, Spill is set equal to the Regulated Spill.

4. If neither Regulated Spill nor Spill are input, they are both set equal to the minRegSpill.

After Release has been calculated, the Regulated spill function may be called a second time if the sum of Release and Spill is less than the Outflow.

The following calculations and evaluations are performed if the function is called for the second time:

1. If either Spill or Regulated Spill are input, a **RiverWare**™ error is posted stating that **RiverWare**™ is unable to allocate the excess flow and the run is aborted.

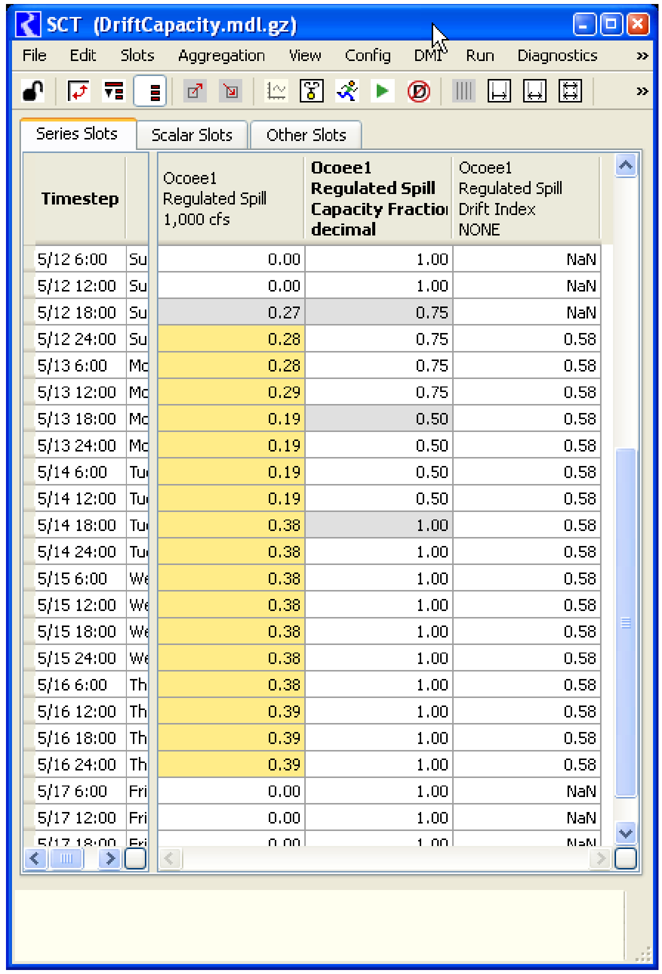
2. Regulated Spill is set equal to the Outflow minus the Release. If Regulated Spill is greater than the maxRegSpill, a **RiverWare**™ error is flagged informing the user that Outflow is greater than the spillway capacities and Release and the run is aborted.

3. Spill is set equal to Regulated Spill.

DRIFT CALCULATIONS:

The drift function is used to calculate Regulated Spill at a specific timestep if it is flagged DRIFT. If the current timestep's Regulated Drift Index is not known, but the previous timestep's Regulated Drift Index is known, the current Regulated Drift Index is set equal to the previous timestep's Regulated Drift Index.

The Drift tables assume that the full spill works is available. Therefore, if there is a Capacity Fraction that is less than 1.0, the Drift calculation (for both regulated and bypass) must be modified.



Timestep		Ocoee1 Regulated Spill 1,000 cfs	Ocoee1 Regulated Spill Capacity Fractio decimal	Ocoee1 Regulated Spill Drift Index NONE
5/12 6:00	Su	0.00	1.00	NaN
5/12 12:00	Su	0.00	1.00	NaN
5/12 18:00	Su	0.27	0.75	NaN
5/12 24:00	Su	0.28	0.75	0.58
5/13 6:00	Mc	0.28	0.75	0.58
5/13 12:00	Mc	0.29	0.75	0.58
5/13 18:00	Mc	0.19	0.50	0.58
5/13 24:00	Mc	0.19	0.50	0.58
5/14 6:00	Tu	0.19	0.50	0.58
5/14 12:00	Tu	0.19	0.50	0.58
5/14 18:00	Tu	0.38	1.00	0.58
5/14 24:00	Tu	0.38	1.00	0.58
5/15 6:00	We	0.38	1.00	0.58
5/15 12:00	We	0.38	1.00	0.58
5/15 18:00	We	0.38	1.00	0.58
5/15 24:00	We	0.38	1.00	0.58
5/16 6:00	Th	0.38	1.00	0.58
5/16 12:00	Th	0.39	1.00	0.58
5/16 18:00	Th	0.39	1.00	0.58
5/16 24:00	Th	0.39	1.00	0.58
5/17 6:00	Fri	0.00	1.00	NaN
5/17 12:00	Fri	0.00	1.00	NaN
5/17 18:00	Fri	0.00	1.00	NaN

With the Drift flag is set, if there is a valid Capacity Fraction[t-1] that is not equal to 1.0, then the Capacity Fraction[t] is set to the previous value, but not overwriting inputs or rule values. This causes the Capacity Fraction to remain throughout the drift operation unless it is changed via a new user input. The screenshot to the right shows a sample run. The Capacity fraction is set to 0.75 on 5/12 18:00 and that value remains until a new value is set via user input on 5/13 18:00. Although this set of inputs may not make physical or operations sense, is shows how the algorithm would perform given the inputs shown.

The current Regulated Drift Index is then used in conjunction with the average Pool Elevation over the current timestep and the Regulated Spill Index Table to obtain the current timestep's Regulated Spill.

If it is the first DRIFT timestep, a Gate Index must be calculated. This is done by using an average of the Pool Elevation at t-2 (when available) and at t-1, the previous regulated spill, the greatest and least spill values, and the Regulated Spill Index Table. If there is a non-zero capacity fraction, the value used for previous regulated spill is adjusted as follows:

$$\text{Regulated Spill}[t-1]_{Adj} = \frac{\text{Regulated Spill}[t-1] - \text{Closed Gate Overflow}[t-1]}{\text{Regulated Spill Capacity Fraction}[t-1]}$$

This computes the spill that would have occurred if all the gates were available. If the previous regulated spill is less than the smallest possible Regulated Spill or greater than the largest possible Regulated Spill (according to the Regulated Spill Index Table), a **RiverWare**™ error is flagged and the run is aborted.

The Regulated Drift Index slot is then set for the current timestep. Finally, the Regulated Spill can be determined from the three dimensional interpolation of the Regulated Spill Index Table using the average Pool Elevation over the timestep and the Regulated Spill Index.

At the end of the algorithm, the computed regulated spill (assuming all the gates are available) is multiplied by the Capacity Fraction [t] to determine spill that will occur with the given capacity fraction. This Regulated Spill is then set on the slot.

17.1.8.5 Regulated and Unregulated

The Regulated and Unregulated method models **Spill** through one controlled, **Regulated Spill**, and one uncontrolled spillway, **Unregulated Spill**. First, the **Unregulated Spill** can not be specified (input or via rules) or a **RiverWare**™ error will abort the run. The **Unregulated Spill** is a function of the average reservoir **Pool Elevation** and takes precedence (i.e. water goes through it first) over other types of outflow (i.e. **Release** or **Turbine Release** or **Regulated Spill**) in the reservoir. When this method is chosen the user category Unregulated Spill Type , Section 17.1.9, will appear.

Second, the user may specify (input or via rules) either

- No slots
- **Spill** or
- **Regulated Spill**

If one is specified and there is excess flow which cannot be met by the (**Turbine**) **Release**, an error will be flagged and the simulation halted.

Slots Specific to this Method

☛ Regulated Spill

Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare**™. If regulated spill is set by the user and the value is greater than the required spill, an error is flagged and the simulation is halted.
Links: Usually not linked. But, it can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☛ Regulated Spill Capacity Fraction

Type: Series Slot
Units: Decimal

Description: The fraction of the Regulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Regulated Spill Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index
Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Regulated Spill Index Table.
Links: Not linkable

☛ Regulated Spill Index Table

Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Regulated Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.
Links: Not linkable

Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column 3.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130
4	550	180
4	600	230

☛ Regulated Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding Max Regulated Spill values
Information:
I/O: Required input
Links: Not linkable

☛ Unregulated Spill

Type: Series Slot
Units: FLOW
Description: spill corresponding to the average Pool Elevation over the timestep
Information:
I/O: Output only
Links: Not linkable

☛ Unregulated Spill Capacity Fraction

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Unregulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 50 ft of a 1000 ft long crest is blocked, the Unregulated Spill Capacity Fraction would be input to 0.95.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Unregulated Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding Unregulated Spill values
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required input
Links: Not linkable

Note: Regulated Spill and Spill are both output slots if neither is input by the user. Only one of these slots, however, can be input for any timestep. When this is the case, the other slot will be an output slot.

The first step in the regulatedPlusUnregSpillCalc method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 17.1.10.2. If not, the minRegSpill is 0.0. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

The Unregulated Spill is then calculated through the steps described in the “Unregulated”. If the Unregulated Spill is input by the user, a **RiverWare**™ error is flagged and the run is aborted.

Release is then checked to see if it has been calculated. If Release is not known, it means that the method is being called for the first time for the particular timestep and the following steps are taken:

1. If both the Spill and Regulated Spill are input (remember setting a DRIFT flag is considered an input), a **RiverWare**™ error is flagged and the run is aborted.

2. If Spill is input and it is greater than the sum of the Unregulated Spill and the maxRegSpill, a **RiverWare**™ error is flagged which states that the requested Spill cannot be met. If the Spill is input and less than the Unregulated Spill plus minRegSpill, a **RiverWare**™ error is flagged. Otherwise, Regulated Spill is calculated as Spill minus Unregulated Spill.

3. If Regulated Spill is input by the user and the DRIFT flag is set, a function is called to perform the drift calculations. A description of the DRIFT function is given in the Regulated Spill section. If Regulated Spill is input and greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted. Otherwise, Spill is set equal to the Regulated Spill plus Unregulated Spill.

4. If neither Regulated Spill nor Spill are input, Regulated Spill is set equal to the minRegSpill and Spill is set equal to Unregulated Spill plus minRegSpill.

After the Release has been calculated, the Regulated and Unregulated function may be called a second time if the sum of Release and Spill is less than the Outflow.

The following calculations are performed if the function is called for the second time:

1. If either Spill or Unregulated Spill are input, an error is posted because there are no free spill variables and **RiverWare**™ is unable to allocate the excess Outflow.

2. The Regulated Spill is calculated using the following equation:

$$\text{Regulated Spill} = \text{Outflow} - \text{Turbine Release} - \text{Unregulated Spill}$$

3. If Regulated Spill is greater than the maxRegSpill, a **RiverWare™** error is posted stating that the Outflow is greater than the spillway capacities and Release and the run is aborted.

4. Spill is calculated as Regulated Spill plus Unregulated Spill.

17.1.8.6 Regulated and Bypass

The Regulated and Bypass method models spill through two regulated spillways called **Regulated Spill** and **Bypass**. The user may specify (input or via rules):

- No slots
- **Spill**
- **Spill** and **Bypass**
- **Spill** and **Regulated Spill**
- **Bypass**
- **Regulated Spill**, or
- **Bypass** and **Regulated Spill**

If all three slots are specified, an error will be issued. Also, if **Spill** is specified and there is excess flow that cannot be met by (**Turbine**) **Release**, a **RiverWare™** error will be flagged and the simulation halted.

The order in which water goes through the various outflow structures depends on what is known. Input/Rules values take precedence, followed by (**Turbine**) **Release**, followed by **Regulated Spill**, and finally by **Bypass**. For example, on a timestep where there is zero (**Turbine**) **Release** and no spill slots are specified, outflows will first go through **Regulated Spill** and any flow greater than max regulated spill will go through **Bypass**.

Slots Specific to this Method

☛ Bypass

Type: Series Slot
Units: FLOW
Description: flow through the Bypass spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare™**. If Bypass is set by the user and the value is greater than the required spill, a **RiverWare™** error is flagged and the simulation is halted.
Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Bypass behaves the same as if it were input.

☛ Bypass Capacity Fraction

Type: Series Slot

Units: DECIMAL
Description: The fraction of the Bypass structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 2 of 8 gates are unavailable, the Bypass Capacity Fraction would be input to 0.75.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ BYPASS DRIFT INDEX
Type: Series Slot
Units: NONE
Description: gate setting index for the Bypass spillway
Information: If the user has set the DRIFT flag on the Bypass slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Bypass Index Table.
Links: Not linkable

☛ BYPASS INDEX TABLE
Type: Table Slot
Units: NOUNITS VS. LENGTH VS. FLOW
Description: Gate Index vs. Pool Elevation vs. Bypass Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Bypass spill slot, this data table must be provided.
Links: Not linkable

☛ BYPASS TABLE
Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding maximum bypass spill values
Information:
I/O: Required input
Links: Not linkable

☛ REGULATED SPILL
Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare**™.

Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☛ Regulated Spill Capacity Fraction

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Regulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Regulated Spill Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index for the Regulated Spill
Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Regulated Spill Index Table.
Links: Not linkable

☛ Regulated Spill Index Table

Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Regulated Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.
Links: Not linkable

Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column 3.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160

Gate Index	Pool Elevation	Spill
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130
4	550	180
4	600	230

☛ Regulated Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding maximum regulated spill values
Information:
I/O: Required input
Links: Not linkable

Note: Spill, Regulated Spill, and Bypass can all be outputs if they are not specified by the user. The user may specify Spill, or either Regulated Spill or Bypass, or both Regulated Spill and Bypass. The slots which are not specified will be output slots.

The first step in the Regulated Spill method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 17.1.10.2. If not, the minRegSpill is 0.0. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

The next step is to obtain the maximum Bypass, maxBypass, by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Bypass Table. Then the Bypass Capacity Fraction is applied as follows:

$$\text{maxBypass} = \text{value from Bypass Table} \times \text{Bypass Capacity Fraction}$$

1. If Spill, Bypass, and Regulated Spill are all input/rules, a **RiverWare**™ Error is posted and the simulation run is aborted.

2. If Spill and Regulated Spill are input/rules by the user, the following steps are performed (Remember: If the DRIFT flag is set on the Regulated Spill slot, Regulated Spill is considered an input.):
   - Drift calculations are performed if the DRIFT flag is set on the Regulated Spill slot. See Regulated Spill for a description of the drift calculations.
   - If Regulated Spill is greater than either Spill or maxRegSpill a **RiverWare**™ error is flagged and the run is aborted.
   - If Regulated Spill is less than minRegSpill, a **RiverWare**™ error is flagged and the run is aborted.
   - Bypass is calculated as Spill minus Regulated Spill.
   - If Bypass is greater than the maxBypass, an error is flagged.
3. If Spill and Bypass are input, the following steps are taken (Remember: If the DRIFT flag is set on the Bypass slot, Bypass is considered an input.):
   - Drift calculations are performed if the DRIFT flag is set on the Bypass slot. The DRIFT calculations are performed in a similar manner to the drift calculations for Regulated Spill which are explained in the Regulated Spill method description.
   - If Bypass is greater than either Spill or the maxBypass, a **RiverWare**™ error is flagged and the simulation is aborted.
   - Regulated Spill is calculated as Spill minus Bypass.
   - If Regulated Spill is either greater than the maximum regulated spill or less than the minimum regulated spill, an error is flagged and the run is aborted.
4. If Spill is input but neither Regulated Spill nor Bypass are input or flagged as DRIFT, the following steps are taken:
   - Regulated Spill is set as the lesser value of either the Spill or the maxRegspill. The Regulated Spill also cannot be less than the minRegSpill.
   - If Regulated Spill is less than Spill, Bypass is calculated as Spill minus Regulated Spill.
   - If Bypass is greater than the maxBypass, a **RiverWare**™ error is flagged and the run is aborted.
   - If Regulated Spill is equal to Spill, Bypass is set equal zero.
5. If Spill is not input, and both Bypass and Regulated Spill are input, the following steps are taken:

- The drift calculations are performed for both Regulated Spill and Bypass if the DRIFT flags have been set. A description of the DRIFT calculations is contained in the Regulated Spill method.
- Regulated Spill and Bypass are checked against the maxRegSpill and maxBypass, respectively. If either Regulated Spill is greater, a **RiverWare**™ error is flagged and the run is aborted.
- Spill is calculated as Regulated Spill plus Bypass.

6. If the DRIFT flag is set on the Regulated Spill, the drift calculations are performed (as described in the Regulated Spill method) to calculate Regulated Spill. The calculated Regulated Spill value is then checked against spillway capacities. If only Regulated Spill is input, the value is checked against the spillway capacity. Spill is set equal to Regulated Spill and Bypass is set to zero if the Regulated Spill is less than or equal to the maxRegSpill. If the Regulated Spill is greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted and the simulation run is aborted.

7. If the DRIFT flag is set on the Bypass, the drift calculations are performed (as described in the Regulated Spill method) to calculate Bypass. The calculated Bypass value is then checked against spillway capacities. If only Bypass is input, the input value is checked against the spillway capacity. Spill is calculated as Bypass plus Regulated Spill. Regulated Spill is set to the minimum regulated spill, if the Bypass is less than the maximum bypass. If the Bypass is greater than the maxBypass, a **RiverWare**™ error is posted and the simulation run is aborted.

8. If no slots are input, Spill is set equal minRegSpill. Bypass is set to zero and Regulated Spill is set to minRegSpill.

After Release has been calculated, the Regulated and Bypass function may be called a second time if the sum of the Release and Spill is less than the Outflow.

The following calculations are performed if the function is called for the second time:

1. If either Spill is input, or both Regulated Spill and Bypass are input, a **RiverWare**™ error is flagged and the simulation run is aborted because there are no free spill variables.

2. If only Regulated Spill is input or flagged as DRIFT, Bypass is recalculated using the following formula:

$$\text{Bypass} = \text{Outflow} - \text{Regulated Spill} - \text{Turbine Release}$$

Bypass is then checked against its spillway capacity and a **RiverWare**™ error is flagged and the simulation is aborted the spillway capacity is exceeded. Spill is calculated as Bypass plus Regulated Spill if the Bypass is less than or equal to maximum allowable bypass.

3. If only Bypass is input or flagged as DRIFT, Regulated Spill is recalculated using the following formula:

$$\text{Regulated Spill} = \text{Outflow} - \text{Bypass} - \text{Turbine Release}$$

The Regulated Spill is then checked against its spillway capacity and minRegSpill. A **RiverWare™** error is posted and the simulation run is aborted if the spillway capacity is exceeded or is less than minimum. Spill is calculated as the sum of Bypass and Regulated Spill.

4. If neither Bypass nor Regulated Spill is input, the following steps are performed:
   - A local variable, excess, is calculated as Outflow minus Turbine Release minus minRegSpill.
   - Regulated Spill is set equal to the lesser value of excess or maxRegSpill but must be greater than minRegSpill.
   - If Regulated Spill is less than the excess, Bypass is calculated as excess minus Regulated Spill.
   - Bypass is checked against its spillway capacity. If Bypass is greater than the maxBypass, a **RiverWare™** error is posted and the simulation run is aborted.
   - Spill is calculated as the Bypass plus the Regulated Spill.
   - If Regulated Spill is equal to the Excess, Bypass is set equal to zero and Spill is set equal to Regulated Spill.

17.1.8.7 Regulated, Bypass and Unregulated

This method models spill through two controlled spillways called **Bypass** and **Regulated Spill** and one uncontrolled spillway called **Unregulated Spill**. The user may not specify (input or via rules) the **Unregulated Spill**. This value is always output and is a function of the average reservoir Pool Elevation. The user may specify (input or rules):

- No slots
- **Spill**
- **Spill** and **Bypass**
- **Spill** and **Regulated Spill**
- **Bypass**
- **Regulated Spill**, or
- **Bypass** and **Regulated Spill**

If **Spill**, **Regulated Spill**, and **Bypass** are specified, an error will be issued. Also, if **Spill** is specified and there is excess flow that cannot be met by (**Turbine**) **Release**, a **RiverWare™** error will be flagged and the simulation halted.

The order in which water goes through the various outflow structures depends on what is known. **Unregulated Spill** takes precedence, followed by input/rules values, followed by (**Turbine**) **Release**, followed by **Regulated Spill**, and finally by **Bypass**. For example, on a timestep where there is zero (**Turbine**) **Release** and no spill slots are specified, outflows will first go through **Unregulated Spill** (computed based on pool elevation), then **Regulated Spill** up to capacity and any excess flows will go through **Bypass**.

SLOTS SPECIFIC TO THIS METHOD

☛ BYPASS

Type: Series Slot
Units: FLOW
Description: flow through the Bypass spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare™**. If Bypass is set by the user and the value is greater than the required spill, a **RiverWare™** error is flagged and the simulation is halted.
Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Bypass behaves the same as if it were input.

☛ BYPASS CAPACITY FRACTION

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Bypass structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 2 of 8 gates are unavailable, the Bypass Capacity Fraction would be input to 0.75.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ BYPASS DRIFT INDEX

Type: Series Slot
Units: NONE
Description: gate setting index for the Bypass spillway
Information: If the user has set the DRIFT flag on the Bypass slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Bypass Index Table.
Links: Not linkable

☛ Bypass Index Table

Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Bypass Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Bypass spill slot, this data table must be provided.
Links: Not linkable

☛ Bypass Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding maximum bypass spill values
Information:
I/O: Required input
Links: Not linkable

☛ Regulated Spill

Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare**™.
Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☛ Regulated Spill Capacity Fraction

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Regulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Regulated Spill Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index for the Regulated Spill

Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if this slot is not set by the user, the gate index is calculated from the Regulated Spill Index Table.
Links: Not linkable

☛ Regulated Spill Index Table

Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Regulated Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.
Links: Not linkable

Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column 3.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130
4	550	180
4	600	230

☛ Regulated Spill Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding maximum regulated spill values
Information:
I/O: Required input
Links: Not linkable

☛ Unregulated Spill

Type: Series Slot
Units: FLOW
Description: spill corresponding to the average Pool Elevation over the timestep
Information:
I/O: Output only
Links: Not linkable

☛ Unregulated Spill Capacity Fraction

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Unregulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 50 ft of a 1000 ft long crest is blocked, the Unregulated Spill Capacity Fraction would be input to 0.95.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Unregulated Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding unregulated spill values
Information:
I/O: Required input
Links: Not linkable

Note: Spill, Regulated Spill, and Bypass may be output slots if they are not specified as input by the user. The user may specify either Spill or Regulated Spill or Bypass, or both Unregulated Spill and Bypass as input. The slots which are not set as input will be output slots.

The first step in the Regulated Spill method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 17.1.10.2. If not, the minRegSpill is 0.0. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \\ \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

The next step is to obtain the maximum Bypass, maxBypass by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Bypass Table. Then the Bypass Capacity Fraction is applied as follows:

$$\text{maxBypass} = \text{value from Bypass Table} \times \text{Bypass Capacity Fraction}$$

The Unregulated Spill is then calculated through the steps described in the “Unregulated”. If the Unregulated Spill is input by the user, a **RiverWare™** error is flagged and the run is aborted.

If Spill, Bypass, and Regulated Spill are input, Spill is overdetermined and a **RiverWare™** error is flagged and the simulation run is aborted.

If Release has not been calculated, the method is executing for the first time in the current timestep and the following steps are taken:

1. If Spill is input by the user and Unregulated Spill is greater than the Spill, a **RiverWare™** error is flagged and the simulation run is aborted.
2. If Spill and Regulated Spill are input the following steps are taken (Remember: If the DRIFT flag is set on the Regulated Spill slot, Regulated Spill is considered an input):
   - Drift calculations are performed if the DRIFT flag is set on the Regulated Spill slot. See Regulated Spill for a description of the drift calculations.
   - If Regulated Spill is greater than either Spill or maxRegSpill, a **RiverWare™** error is flagged and the run is aborted.
   - If Regulated Spill is less than minRegSpill, a **RiverWare™** error is flagged and the run is aborted.
   - Bypass is calculated as Spill minus Regulated Spill minus Unregulated Spill.
   - If Bypass is either greater than the maxBypass or less than zero, an error is flagged.
3. If Spill and Bypass are input, the following steps are taken (Remember: If the DRIFT flag is set on the Bypass slot, Bypass is considered an input.):
   - Drift calculations are performed if the DRIFT flag is set on the Bypass slot. The DRIFT calculations are performed in a similar manner to the drift calculations for Regulated Spill, which are explained in the Regulated Spill method description.
   - If Bypass is greater than either Spill or the maxBypass, a **RiverWare™** error is flagged and the run is aborted.
   - Regulated Spill is calculated as Spill minus Bypass minus Unregulated Spill.

- If Regulated Spill is either greater than the maxRegSpill or less than the minRegSpill, an error is flagged and the run is aborted.

4. If Spill is input by the user and neither Bypass nor Regulated Spill are input, the following steps are taken:
   - Regulated Spill is set as the lesser value of either the Spill minus Unregulated Spill or the maxRegSpill. The Regulated Spill also cannot be less than the minRegSpill.
   - If Regulated Spill is less than Spill minus Unregulated Spill, Bypass is set equal to Spill minus Regulated Spill minus Unregulated Spill.
   - If Bypass is greater than the maxBypass, a **RiverWare**™ error is flagged and the run is aborted.
   - If Regulated Spill is equal to Spill minus Unregulated Spill, Bypass is set equal to zero.

5. If Spill is not input, but both Bypass and Regulated Spill are input, the following steps are taken:
   - The drift calculations are performed for both Regulated Spill and Bypass if the DRIFT flags have been set. A description of the DRIFT calculations is contained in the Regulated Spill method.
   - Regulated Spill and Bypass are checked against the maxRegSpill and maxBypass, respectively. If either Regulated Spill is greater, a **RiverWare**™ error is flagged and the run is aborted. If Regulated Spill is less than the minRegSpill, an error is issued.
   - Spill is calculated as the sum of Regulated Spill, Bypass, and Unregulated Spill.

6. If the DRIFT flag is set on the Regulated Spill, the drift calculations are performed (as described in the Regulated Spill method) to calculate Regulated Spill. The calculated Regulated Spill Value is then checked against spillway capacities. If only Regulated Spill is input, the value is checked against the spillway capacity. Spill is calculated as Regulated Spill plus Unregulated Spill and Bypass is set to zero, if the Regulated Spill is less than the maxRegSpill. If the Regulated Spill is greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted and the simulation run is aborted.

7. If the DRIFT flag is set on the Bypass, the drift calculations are performed (as discussed in the Regulated Spill method) to calculated the Bypass. The calculated Bypass value is then checked against spillway capacities. If only Bypass is input, the input value is checked against the spillway capacity. Spill is calculated as Bypass plus Unregulated Spill and Regulated Spill is set to the minRegSpill, if the Bypass is less than the maxBypass. If the Bypass is greater than the maxBypass, a **RiverWare**™ error is posted and the simulation run is aborted.

8. If no slots are input, Spill is set equal to Unregulated Spill plus minRegSpill. Bypass is set to zero and Regulated Spill is set to minRegSpill.

After Release has been calculated, the Regulated, Bypass and Unregulated function may be called a second time if the sum of the Release and Spill is less than the Outflow.

The following calculations are performed if the function is called for the second time:

1. If either Spill is input or both Regulated Spill and Bypass are input, a **RiverWare**™ error is flagged and the simulation run is aborted because there are no free spill variables.

2. If only Regulated Spill is input or flagged as DRIFT, Bypass is recalculated using the following formula:

$$\text{Bypass} = \text{Outflow} - \text{Regulated Spill} - \text{Unregulated Spill} - \text{Turbine Release}$$

The Bypass is then checked against its spillway capacity. A **RiverWare**™ error is posted and the simulation run is aborted if the spillway capacity is exceeded. Spill is calculated as the sum of Bypass, Unregulated Spill, and Regulated Spill if the Bypass is less than or equal to the maxBypass.

3. If only Bypass is input, Regulated Spill is recalculating the following formula:

$$\text{Regulated Spill} = \text{Outflow} - \text{Bypass} - \text{Unregulated Spill} - \text{Turbine Release}$$

The Regulated Spill is then checked against its spillway capacity and minRegSpill. A **RiverWare**™ error is posted and the simulation run is aborted if the spillway capacity is exceeded or is less than minimum. Spill is calculated as the sum of Bypass, Regulated Spill, and Unregulated Spill if the Regulated Spill is less than or equal to the maximum allowable regulated spill.

4. If neither Regulated Spill nor Bypass are input, the following steps are performed:
   - A local variable, excess, is calculated as Outflow minus Unregulated Spill minus Turbine Release minus minimum regulated spill.
   - Regulated Spill is set equal to the lesser value of excess or maxRegSpill but must be greater than minRegSpill.
   - If Regulated Spill is less than excess, Bypass is calculated as excess minus Regulated Spill.
   - Bypass is checked against its spillway capacity. If Bypass is greater than the maxBypass, a **RiverWare**™ error is posted and the simulation run is aborted.
   - Spill is set equal to the sum of Bypass, Regulated Spill, and Unregulated Spill.

- If Regulated Spill is equal to Excess, Bypass is set equal to zero and Spill is set equal to the sum of Regulated Spill and Unregulated Spill.

17.1.8.8 Bypass, Regulated and Unregulated

Note: This user method is the similar to the **Regulated, Bypass and Unregulated** method but switches the order of the **Bypass** and **Regulated Spill** outlet works. This method is preferable in institutional cases where the term "**Bypass**" is favored over the term "**Regulated Spill**". Other than the order reversal, the functionality is similar to the Regulated, Bypass and Unregulated method.

This method models spill through two controlled spillways called **Bypass** and **Regulated Spill** and one uncontrolled spillway called **Unregulated Spill**. The user may not specify (input or via rules) the **Unregulated Spill**. This value is always output and is a function of the average reservoir **Pool Elevation**. The user may specify (input or rules):

- No slots
- **Spill**
- **Spill** and **Bypass**
- **Spill** and **Regulated Spill**
- **Bypass**
- **Regulated Spill**, or
- **Bypass** and **Regulated Spill**

If **Spill**, **Regulated Spill**, and **Bypass** are specified, an error will be issued. Also, if **Spill** is specified and there is excess flow that cannot be met by (**Turbine**) **Release**, a **RiverWare™** error will be flagged and the simulation halted.

The order in which water will go through the various outflow structures depends on what is known. **Unregulated Spill** takes precedence, followed by input/rules values, followed by (**Turbine**) **Release**, followed by **Bypass**, and finally by **Regulated Spill**. For example, on a timestep where there is zero (**Turbine**) **Release** and no spill slots are specified, outflows will first go through **Unregulated Spill** (required based on pool elevation), then **Bypass** up to capacity and any excess flows will go through **Regulated Spill**.

Please see the **regPlusBypassPlusUnregSpill** method for a description of the slots particular to this method and the algorithm of this method. The algorithm for this method is only different in that **Bypass** takes precedence over **Regulated Spill**.

17.1.9 Unregulated Spill Type

This category is only visible when a method using **Unregulated Spill** is chosen. The three Unregulated Spill Types are Bare Crest Only, Two Unregulated Flow, and Three Unregulated Flows.

17.1.9.1 Bare Crest Only

The Bare Crest Only method is the default method in the Unregulated Spill Type Category. The method assumes an unobstructed spillway where the flow over the spillway is a function of the **Unregulated Spill Table**. There are no slots specifically associated with this method.

17.1.9.2 Two Unregulated Flows

When the **Two Unregulated Flows** method is selected, flow over the spillway is a function of the **Unreg Flow 2 Spill Table.** If the pool elevation meets or exceeds the **Unreg Flow 2 Failure Elevation**, flow becomes a function of the **Unregulated Spill Table.**

Note: This method originally was called **Flashboards** but was renamed to be more general. (Flashboards are wooden boards installed in the unregulated spillway so that the reservoir may store more water than what the spillways themselves would allow.)

☛ Unreg Flow 2 Avail and Failure Time

Type: Agg Series Slot
Units: FRACTION
Description: Availability of Unreg Flow 2 Spill Table, fraction of timestep when Unreg Flow 2 Spill Table is in use.
Information:
I/O: Optional: Availability may be input by user or set by a rule. Failure time is output only
Links: Not linkable

☛ Unreg Flow 2 Failure Elevation

Type: Table Slot
Units: LENGTH
Description: Pool Elevation at which Unreg Flow 2 Spill Table is no longer used.
Information:
I/O: Required Input
Links: Not linkable

☛ Unreg Flow 2 Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW

Description: Pool Elevation vs. corresponding unregulated flow 2 spill values.
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required Input
Links: Not linkable

METHOD DETAILS

To determine the spill tables to use during a timestep based on availability of Unreg Flow 2, the following three possibilities are checked:

1. Availability is input or set by a rule: no change from Unreg Flow 2 to Unregulated Spill over timestep, calculate spill based on availability.

2. Availability is 0: no change from Unreg Flow 2 to Unregulated Spill, calculate spill based on Unregulated Spill Table.

3. Availability is greater than 0, check if failure from Unreg Flow 2 to Unregulated Spill occurs during the timestep.

During the third case, the spill over the timestep is calculated using the respective table and multiplied by the availability to find the total spill over the timestep. The total spill is used to predict the pool elevation at the end of the timestep. The calculated pool elevation is compared to the Unreg Flow 2 Failure Elevation. If a failure from Unreg Flow 2 to Unregulated Spill is found to occur any time during the timestep, the failure time is recorded and the ending pool elevation is re-calculated to account for the change in spill due to the change from Unreg Flow 2 to Unregulated Spill during the timestep.

Note: If a failure from Unreg Flow 2 to Unregulated Spill occurs during a dispatch, the first time of this failure is used for the remainder of the dispatch.

The time of failure during the timestep is used to determine what portion of the timestep needs interpolation from each of the two spill tables.

17.1.9.3 Three Unregulated Flows

When the **Three Unregulated Flows** is selected, flow over the spillway is a function of the **Unreg Flow 3 Spill Table.** If the pool elevation meets or exceeds the **Unreg Flow 3 Failure Elevation**, flow becomes a function of the **Unreg Flow 2 Spill Table.** If the pool elevation meets or exceeds the **Unreg Flow 2 Failure Elevation**, flow becomes a function of the **Unreg Flow Spill Table.** To summarize, Unreg Flow 3 fails first, then Unreg Flow 2 fails next as the pool rises. Therefore, the Unreg Flow 2 Failure Elevation should be higher than the Unreg Flow 3 Failure Elevation.

Note: This method was originally called the **Flashboards and Superboards** method but was renamed to be more general. Flashboards and superboards are wooden boards installed in the unregulated spillway so that the reservoir may store more water than what the spillways themselves would allow. The superboards can only be installed if the flashboards are in place.

☛ Unreg Flow Avail and Failure Time

Type: Agg Series Slot
Units: FRACTION
Description: Availability and failure time of Unreg flow 2 and 3 Spill tables.
Information:
I/O: Optional: Availability may be input by user or set by a rule. Failure time is output only
Links: Not linkable

☛ Unreg Flow 3 Failure Elevation

Type: Table Slot
Units: LENGTH
Description: Pool Elevation at which Unreg Flow 3 Spill Table is no longer used.
Information: The value in this slot should be lower than the value in the Unreg Flow 2 Failure Elevation.
I/O: Required Input
Links: Not linkable

☛ Unreg Flow 3 Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding unregulated flow 3 spill values.
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required Input
Links: Not linkable

☛ Unreg Flow 2 Failure Elevation

Type: Table Slot
Units: LENGTH
Description: Pool Elevation at which Unreg Flow 2 Spill Table is no longer used.
Information: The value in this slot should be higher than the value in the Unreg Flow 2 Failure Elevation.
I/O: Required Input
Links: Not linkable

☛ Unreg Flow 2 Spill Table

Type: Table Slot
Units: Length vs. Flow
Description: Pool Elevation vs. corresponding unregulated flow 2 spill values.
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required Input
Links: Not linkable

Method Details

To determine the spill tables to use during a timestep based on availability of Unreg Spill 2 and Unreg Spill 3, the following four possibilities are checked:

1. Both availabilities are input or set by a rule: no change in unregulated spill type over timestep, calculate spill based on availability.
2. One availability is input or set by a rule: Error, both or none must be input.
3. Availabilities are 0. No change in unregulated spill type over timestep. Calculate spill based on Unregulated Spill Table.
4. At least one availability is greater than 0, check for failure in unregulated spill type.

During the fourth case the spill over the timestep is calculated using the respective table and multiplied by the availability to find the total spill over the timestep. The total spill is used to project the pool elevation at the end of the timestep. The calculated pool elevation is compared to the failure elevations of Unreg Flow 2 or Unreg Flow 3. If failure in unregulated spill type is found to occur any time during the timestep, this failure time is recorded and the ending pool elevation is recalculated to account for the change in spill due to the change in unregulated spill type during the timestep.

Note: If a change in unregulated spill type occurs during a dispatch, the first time of this change is used for the remainder of the dispatch.

The time at which change in unregulated spill type occurred during the timestep is used to determine what portion of the timestep needs interpolation from each of the three spill tables.

17.1.10 Regulated Spill Overflow

The category, **Regulated Spill Overflow,** is added if one of the following "regulated" spill methods is selected:

- **Regulated Spill**
- **Regulated and Unregulated**
- **Regulated and Bypass**
- **Regulated, Bypass and Unregulated**
- **Bypass, Regulated and Unregulated**

17.1.10.1 None

This is the default, no-action method.

17.1.10.2 Closed Gate Overflow

This method models the uncontrolled flow over a closed regulated spill gate. This functionality uses the **Regulated Spill Capacity Fraction** to compute the default amount of spillway that is overtopped.

This functionality only applies to **Regulated Spill**, not **Bypass**.

THE FOLLOWING SLOTS WILL BE ADDED:

☛ **CLOSED GATE OVERFLOW**

Type: Series Slot
Units: FLOW
Description: Uncontrolled portion of the Regulated spill that overtops the gates.
Information: This value is computed by the regulated spill method as the value found on the **Closed Gate Overflow Table** multiplied by the **Closed Gate Overflow Capacity Fraction**
I/O: Output only
Links: Not linkable

☛ **CLOSED GATE OVERFLOW TABLE**

Type: Table Slot
Units: LENGTH VS FLOW
Description: Pool Elevation vs unregulated flow.
Information: This table is used to specify the rating curve for uncontrolled flow over the closed Regulated Spill gates. The values should be input to the table as though every Regulated Spill gate is closed. The tables would start with zero flow at or just below the top of the closed gates.
I/O: Input Only
Links: Not available

☛ Closed Gate Overflow Capacity Fraction

Type: Series Slot
Units: FRACTION
Description: The fraction of the closed gate overflow that is available.
Information: The value must be between 0.0 and 1.0, inclusive. If not input or set by a rule, it defaults to (1 - **Regulated Spill Capacity Fraction**). Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be set to 0.875 and the Closed Gate Overflow Capacity Fraction would default to 0.125.
I/O: Input, set by a rule, or output
Links: Not linkable

Method Details

If not input or set by a rule, **Closed Gate Overflow Capacity Fraction** defaults to (1 - **Regulated Spill Capacity Fraction**). This default indicates that the overflow only happens over gates that are closed. Otherwise, the user can specify the **Closed Gate Overflow Capacity Fraction** slot to say how much of the overflow structure is available.

When the reservoir is below the top of the gates, there is no **Closed Gate Overflow**. But once the reservoir is above the top of one or more closed gates, there is **Closed Gate Overflow**. The computation of this overflow is similar to the unregulated spill computation:

1. A temporary variable called “initHW” is created to represent the Pool Elevation at the beginning of the timestep. Likewise, “endHW” is created to represent the Pool Elevation at the end of the timestep (If the Pool Elevation at the end of the timestep is not known, endHW is set equal to the Pool Elevation at the beginning of the timestep.)

2. The “Closed Gate Overflow Crest” is found from the Closed Gate Overflow Table. It is the Pool Elevation that corresponds to an overflow of zero.

3. If both initHW and endHW are less than or equal to the Closed Gate Overflow Crest, Closed Gate Overflow is set equal to zero.

4. If both initHW and endHW are greater than the Closed Gate Overflow Crest, the average Pool Elevation is used to determine the Closed Gate Overflow from the Closed Gate Overflow Table.

$$\text{Closed Gate Overflow} = \text{Value from table} \times \text{Closed Gate Overflow Capacity Fraction}$$

5. If either initHW or endHW is greater than the Closed Gate Overflow Crest and the other is lower than the crest, the following evaluations and computations are performed:

$$\text{maxHW} = \text{the greater of initHW and endHW}$$

$$\text{minHW} = \text{the lesser of initHW and endHW}$$

$$\text{avgHW} = \frac{\text{maxHW} + \text{Closed Gate Overflow Crest}}{2}$$

$$\text{overflow fraction} = \frac{\text{maxHW} - \text{Closed Gate Overflow Crest}}{\text{maxHW} - \text{minHW}}$$

where:

maxHW = the maximum value of Pool Elevation over the timestep.

minHW = the minimum value of Pool Elevation over the timestep.

avgHW = the average Pool Elevation causing overflow over the timestep.

overflow fraction = corresponds to the fraction of the timestep during which overflow occurs.

A temporary variable called “temp overflow” is obtained from the linear interpolation of the Closed Gate Overflow Table using avgHW. Closed Gate Overflow is then calculated as:

$$\text{Closed Gate Overflow} = \text{overflow fraction} \times \text{temp overflow} \times \text{Closed Gate Overflow Capacity Fraction}$$

When allocating spills to various structures, the **Closed Gate Overflow** must occur at the same time as unregulated spills (i.e. before regulated or bypass). Then any remaining outflow can go through the regulated and or bypass spill structures. Therefore, the minimum regulated spill is computed as follows:

$$\text{Min Regulated Spill} = \text{value from Closed Gate Overflow Table} \times \text{Closed Gate Overflow Capacity Fraction}$$

Also, the **Closed Gate Overflow** is set equal to the Min Regulated Spill.

The functionality assumes that water is either flowing through the gate or over topping it, but not both. The method assumes that the Gate Overflow Table is fixed, that is the elevations in the table do not change. Thus if you had a gate stuck with 1/2 ft open at the bottom and there was still water going over the top, the table (which assumes the gate is closed) would be an incorrect rating.

17.1.11 Input Outflow Adjustment

This method category is only available if a method is selected in the Spill Calculation category. Its purpose is to adjust input Outflow values if they violate a physical constraint.

17.1.11.1 None

This is the default method. It performs no calculations and there are no slots associated with it. The Outflow values will not be adjusted if this method is selected.

17.1.11.2 Reduce Input Outflow

This method is used to reduce the input Outflow value whenever it exceeds the maximum reservoir outflow (due to outlet works capacity).

SLOTS SPECIFIC TO THIS METHOD

☛ **REQUESTED OUTFLOW**

Type: Series Slot
Units: FLOW
Description: The Outflow value before being adjusted
Information: This slot is available so that the user can see when an Outflow value is adjusted. The value in this slot is the outflow value before being adjusted. A value exists in this slot only if the Outflow value is adjusted.
I/O: Output only
Links: Not linkable

If the Outflow slot value is greater than the maximum reservoir outflow, this method saves the Outflow value in the Requested Outflow slot. Then, the Maximum Capacity flag is set on the Outflow slot. The reservoir is then forced to re-dispatch with the Outflow set to Max Capacity (instead of the original, input value). When the reservoir solves the second time, it computes the maximum reservoir outflow and sets this value on the Outflow slot. The Maximum Capacity flag remains on the Outflow slot for the timestep in question (and will be saved with the model file).

17.1.11.3 Allow Excess Specified Outflows

This method allows input Outflows that exceed the maximum reservoir outflow (due to outlet capacity). Because the excess is above the maximum possible for the (Turbine) Release and Spill slots, it will not be classified as either. Instead, the excess is stored on a separate series slot for reporting or tracking.

SLOTS ASSOCIATED WITH THIS METHOD:

☛ OUTFLOW EXCEEDING MAX

Type: Series Slot
Units: FLOW
Description: The portion of the input Outflow that exceeds the sum of the Spill and (Turbine) Release.
Information: This slot tracks the amount that does not fit through the Release and Spill structures.
I/O: Output Only
Links: Not Linkable

METHOD DETAILS:

Toward the end of each dispatch method, if Outflow is greater than the sum of Spill and (Turbine) Release, the Spill method is executed again to redistribute the Outflows to the appropriate spill structures. Within the Spill method, if there is still **no** room for the specified Outflow, the selected method in the **Input Outflow Adjustment** category is executed. When the **Allow Excess Specified Outflows** method is selected, it does the following:

If the Outflow does not have an input flag (I or Z), then the method exits and issues an error that there are excess outflows.

If the Outflow is input (I or Z flag), the method computes the difference between the specified Outflow and the maximum Outflow (i.e. Turbine Release + Spill). This excess outflow is then set on the **Outflow Exceeding Max** slot.

$$\text{Outflow Exceeding Max}[\,] = \text{Outflow}[\,] - ((\text{Turbine})\ \text{Release}[\,] + \text{Spill}[\,])$$

The method then exits successfully and returns to the Spill method and then the dispatch method. The dispatch method sets the spill and mass balance slots.

17.1.12 Future Value

The methods in this category are used to determine the future value of the energy that would have been generated by the water that was lost through the spillway.

17.1.12.1 None

None is the default method for the Future Value category. No calculations are performed by this method. There are no slots specifically associated with this method.

17.1.12.2 Cumulative Storage Value Table

Slots Specific to this Method

☛ Marginal Storage Value Table

Type: Table
Units: Volume vs. $per Energy
Description: Storage versus marginal value per unit energy
Information: This table should be increasing in storage, and usually decreasing in marginal value.
I/O: Required input
Links: Not linkable

☛ Spill Cost

Type: Series Slot
Units: $
Description: Future cost of energy lost due to spilled water
Information:
I/O: Output only
Links: May be linked to the Spill Cost slot on the Thermal Object.

☛ Future Value of Used Energy

Type: Series Slot
Units: $
Description: Future value of energy used in the current timestep
Information:
I/O: Output only
Links: May be linked to Future Value of Used Energy Slot on the Thermal Object.

☛ Anticipated Storage

Type: Series

Units: VOLUME
Description: The combination of Storage in the reservoir at the given timestep plus any flow (converted volume) that is in transit to the reservoir
Information: This slot represents the storage including any lagged flows that are already in a linked upstream reach, which will reach the reservoir at a later timestep. If there are no lagged reaches between this reservoir and the next upstream reservoir, Anticipated Storage will equal Storage. It is this storage value that will be used to calculate Cumulative Storage Value.
I/O: Output only
Links: May be linked

☛ CUMULATIVE STORAGE VALUE

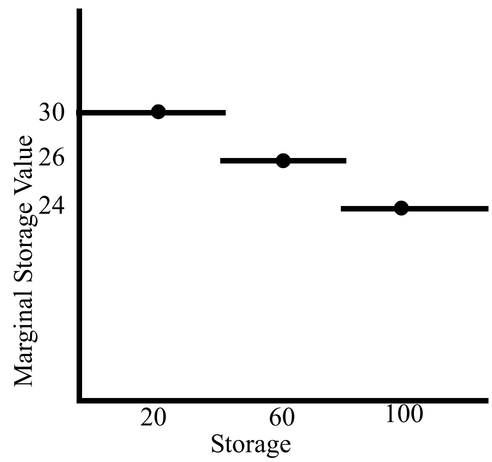
Type: Series Slot
Units: $
Description: Represents the future energy value of the current Anticipated Storage
Information:
I/O: Output only
Links: May be linked to the Total Cumulative Storage Value Slot on the Thermal Object

☛ CUMULATIVE STORAGE VALUE TABLE

Type: Table
Units: VOLUME VS. $
Description: Anticipated Storage and cumulative value used to calculate the Cumulative Storage Value as a function of Anticipated Storage
Information: This table should be increasing in storage and usually increasing in cumulative storage value.
I/O: Required Input either by the user or automatically generated by **RiverWare**™ if the Cumulative Storage Value Table Automation method is selected.
Links: Not linkable

This method uses the Marginal Storage Value Table and the calculated Spill and Turbine Release to compute the Spill Cost and Future Value of Used Energy. It then uses the Cumulative Storage Value Table and the calculated Anticipated Storage to compute the Cumulative Storage Value.

The correct marginal value is found from the current storage in the reservoir. If the current storage is less than the midpoint between the first and second storage table values, the first marginal value is used. The second marginal value is used for a current storage above that midpoint to the midpoint between the second and third storage table values. The last marginal value is used for any current storage above the midpoint between the second-to-last and the last storage table value. An example is shown in Table 8 on page 508



Storage	Marginal Value
20	30
60	26
100	24

Table: 8 Marginal Value Table

Assume that the current storage is 39. This method would use 30 as the marginal value for use in the next computation. Assume that the current storage is 41. In this case the method would use 26 as the marginal value for use in the next computation.

Use of a table in this fashion is unique to this method.

Spill Cost is computed by the following equation:

$$\text{Spill Cost} = \text{Spill} \times \text{Marginal Storage Value} \times \text{Timestep Length}$$

Future Value of Used Energy is computed by the following equation:

$$\text{Future Value of Used Energy} = \text{Turbine Release} \times \text{Marginal Storage Value} \times \text{Timestep Length}$$

The Cumulative Storage Value computation begins by first calculating Anticipated Storage. This is the sum of the reservoir Storage plus any flow already in transit to the reservoir in an upstream lagged reach. For example, assume a reservoir's Inflow slot is linked to a reach with a 3-hour lag time. In an hourly run, the reservoir's Anticipated Storage would be calculated as:

$$\text{Reservoir.Anticipated Storage} = \text{Reservoir.Storage} + (\text{Reach.Inflow(-2)} + \text{Reach.Inflow(-1)} + \text{Reach.Inflow}) \times TimestepLength$$

If there are no lagged reaches between the reservoir and the next upstream reservoir, then Anticipated Storage will simply equal Storage.

The Cumulative Storage Value is then computed by interpolating from the Cumulative Storage Value Table using the calculated Anticipated Storage value.

17.1.13 Cumulative Storage Value Table Automation

This category allows the **RiverWare**™ simulation to automate the creation of the Cumulative Storage Value Table. This category is only visible if Cumulative Storage Value Table is selected in the Future Value category.

17.1.13.1 None

If this method is selected, no automation will be performed and the user must enter the data into the Cumulative Storage Value Table.

17.1.13.2 Marginal Value to Table

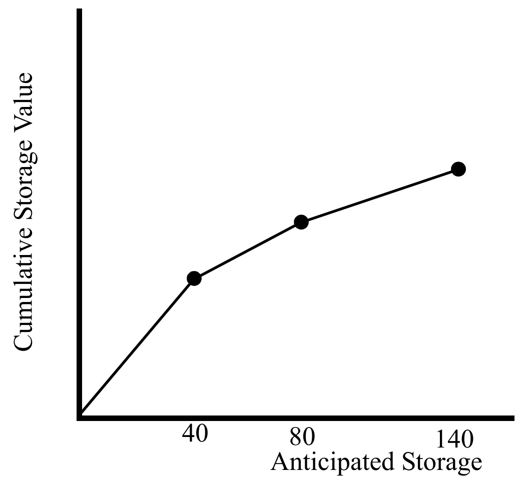
If this method is selected, the Marginal Storage Value table will be used as the source for the generation of the Cumulative Storage Value Table. This is the only calculation associated with this method. There are no slots associated specifically with this method.

This method uses information from the simulation slot Marginal Storage Value Table to generate a Cumulative Storage Value Table. The cumulative storage value can be thought of as the summation of the marginal storage values from a storage of 0 to the current storage. Therefore, the automation method finds the same midpoint values used by the simulation Future Value Calc method, and uses those points in the table.

If only one value exists in the Marginal Storage Value Table, then only two entries will exist in the Cumulative Storage Value Table. The two entries will be 0, and midway between the value in the Marginal Storage Value Table, and the maximum value set on the Storage slot. If more than one value exists, three or more points will result. An example is shown below of a Marginal Value Table and the resulting Cumulative Storage Value Table with a graph of the Cumulative Storage Value data.

Storage	Marginal Value
20	30
60	26
100	24

Table: 9 Marginal Value Table



Anticipated Storage	Cumulative Value
40	1200
80	2240
140	3680

Table: 10 Cumulative Storage Value Table

17.1.14 Ramping

This category allows you to model the cost of turbine ramping, the cost of changing the turbine release from one timestep to the next.

17.1.14.1 None

This is the default method for this category. No new slots are instantiated, and no calculations are performed.

17.1.14.2 Track Ramping

In this method the user assigns a unit cost to turbine ramping, a cost per unit change in flow from one timestep to the next. The method then calculates the change in turbine release and the associated ramping cost. The unit cost is the same for ramping up and ramping down.

SLOTS SPECIFIC TO THIS METHOD

☛ **RAMPING COST**

Type: Series
Units: VALUE
Description: The total cost of turbine ramping for the timestep.
Information: This is the Unit Ramping Cost multiplied by the change in turbine release from one timestep to the net.
I/O: Output only
Links: May be linked

☛ **TURBINE DECREASE**

Type: Series
Units: FLOW
Description: The difference between Turbine Release at the previous timestep and the current timestep when Turbine Release decreases from the previous timestep
Information: If Turbine Release at the current timestep is greater than at the previous timestep, the value is zero.
I/O: Output only
Links: May be linked

☛ **TURBINE INCREASE**

Type: Series
Units: FLOW
Description: The difference between Turbine Release at the current timestep and the previous timestep when Turbine Release increases from the previous timestep

Information: If Turbine Release at the current timestep is less than at the previous timestep, the value is zero.
I/O: Output only
Links: May be linked

UNIT RAMPING COST

Type: Table 1x1
Units: VALUE PER FLOW
Description: The cost per unit change in Turbine Release from one timestep to the next
Information: There is a single value for Unit Ramping Cost (i.e. the same unit cost for ramping up and ramping down)
I/O: Optional input; if not input or negative, defaults to zero
Links: Not linkable

The method first checks for a value in the Unit Ramping Cost table slot. If there is no input value, or if the value is negative, it sets the value to zero. Then it checks if Turbine Release is valid for the current timestep and the previous timestep. If Turbine Release at the current timestep is greater than or equal to the previous timestep then the following values are set:

$$\text{Turbine Increase} = \text{Turbine Release} - \text{Turbine Release}(-1)$$

$$\text{Turbine Decrease} = 0$$

$$\text{Ramping Cost} = \text{Unit Ramping Cost} \times \text{Turbine Increase}$$

Otherwise:

$$\text{Turbine Decrease} = \text{Turbine Release}(-1) - \text{Turbine Release}$$

$$\text{Turbine Increase} = 0$$

$$\text{Ramping Cost} = \text{Unit Ramping Cost} \times \text{Turbine Decrease}$$

If Turbine Release is not valid at both the current and previous timesteps, then Turbine Increase, Turbine Decrease and Ramping Cost will all display NaN.

17.1.15 Hydrologic Inflow

The Hydrologic Inflow category allows **RiverWare**™ to accommodate inflows to a Reservoir that are not part of the main channel and/or are not gauged. The user methods in this category may be used to initialize the Hydrologic Inflow slot if it is required by the user. If the Hydrologic Inflow slot has been initialized, it is figured into the mass balance when the object dispatches.

17.1.15.1 None

None is the default method for the Hydrologic Inflow category. No calculations are performed by this method. There are no slots specifically associated with this method. If this method is selected, the Hydrologic Inflow slot is not initialized so it is no included in the mass balance.

17.1.15.2 Solve Hydrologic Inflow

The Solve Hydrologic Inflow method should be used when the user wishes **RiverWare**™ to calculate Hydrologic Inflow. Hydrologic Inflow will be solved for when the Reservoir dispatches. **RiverWare**™ will not overwrite any user input values for Hydrologic Inflow. Hydrologic Inflow is only solved for when a value is not input.

SLOTS SPECIFIC TO THIS METHOD

☛ **HYDROLOGIC INFLOW**

Type: Series
Units: FLOW
Description: flow into the reservoir that is not gauged and/or does not enter through the main channel.
Information:
I/O: Optional; calculated if not input.
Links: Usually input or calculated but could be linked to the Outflow of any object or any other series slot.

☛ **HYDROLOGIC INFLOW ADJUST**

Type: Series
Units: FLOW
Description: optional adjustment that can be made to the calculated Hydrologic Inflow
Information:
I/O: Optional; set to zero if not input by the user
Links: Not linkable

☛ HYDROLOGIC INFLOW NET

Type: Series
Units: FLOW
Description: sum of Hydrologic Inflow and Hydrologic Inflow Adjust
Information:
I/O: Output only
Links: Not linkable

The following steps are performed in the Solve Hydrologic Inflow method.

First, if Hydrologic Inflow Adjust is not set by the user, it is set equal to zero by **RiverWare**™. Then if Hydrologic Inflow is not set by the user, it is calculated in the dispatch method (see Dispatch Methods for detailed explanation). Hydrologic Inflow Net is calculated as the sum of Hydrologic Inflow and Hydrologic Inflow Adjust.

17.1.15.3 Input Hydrologic Inflow

The Input Hydrologic Inflow method should be used when the user wishes either to input the values of Hydrologic Inflow or have the values default to zero. **RiverWare**™ will not overwrite any user input values.

SLOTS SPECIFIC TO THIS METHOD

☛ HYDROLOGIC INFLOW

Type: Series
Units: FLOW
Description: flow into the reservoir that is not gauged and/or does not enter through the main channel.
Information:
I/O: Optional; defaults to zero if not input.
Links: Usually input or calculated but could be linked to the Outflow of any object or any other series slot.

☛ HYDROLOGIC INFLOW ADJUST

Type: Series
Units: FLOW
Description: optional adjustment that can be made to the calculated Hydrologic Inflow
Information:
I/O: Optional; set to zero if not input by the user.
Links: Not linkable

☛ HYDROLOGIC INFLOW NET

Type: Series

Units: FLOW
Description: sum of hydrologic Inflow and Hydrologic Inflow Adjust
Information:
I/O: Output only
Links: Not linkable

The algorithm used for this method is very simple. If Hydrologic Inflow is not input by the user, it is set equal to zero. Hydrologic Inflow Net is calculated as the sum of Hydrologic Inflow and Hydrologic Inflow Adjust.

17.1.15.4 Hydrologic Inflow and Loss

The Hydrologic Inflow and Loss method should be used when the user wishes to have negative inflows taken into account as unidentified losses.

Slots Specific to this Method

☛ Hydrologic Inflow

Type: Series
Units: FLOW
Description: flow into the reservoir that is not gauged and/or does not enter through the main channel.
Information:
I/O: Optional; defaults to zero if not input.
Links: Usually input or calculated but could be linked to the Outflow of any object or any other series slot.

☛ Hydrologic Inflow Adjust

Type: Series
Units: FLOW
Description: optional adjustment that can be made to the calculated Hydrologic Inflow
Information:
I/O: Optional; the Hydrologic Inflow Adjust may either be input by the user or set to zero if it is not input. If the Inflow is negative, it is added to Hydrologic Inflow Adjust.
Links: Not linkable

☛ Hydrologic Inflow Net

Type: Series
Units: FLOW
Description: sum of Hydrologic Inflow and Hydrologic Inflow Adjust
Information:
I/O: Output only

Links: Not linkable

This method calculates Hydrologic Inflow using the following process.

If the user has selected this method and negative inflows occur, this method will be called from the dispatch method. The Inflow (which in this case is a negative number) will be added to the Hydrologic Inflow Adjust. Then the Inflow is set equal to zero. Otherwise, this method behaves like the Input Hydrologic Inflow method.

17.1.15.5 Forecast Hydrologic Inflow

This method is used to forecast the hydrologic inflow based on known inflow values. When this method is selected, the Generate Forecast Hydrology category becomes visible. Within this category, methods are available to generate the hydrologic inflow forecast.

17.1.16 Generate Forecast Hydrology

This category contains methods that forecast the hydrologic inflow based on known inflow values. The user inputs the historical inflows to the reservoir and the methods adjust those values to represent a forecast. The methods in this category execute at the beginning of each timestep.

17.1.16.1 None

This method is the default for this category. It will result in an error if it is selected and a run is started.

17.1.16.2 Geometric Recession

On each timestep in the forecast period, this method will adjust the inflow hydrographs. If the Forecast Period method is selected for the Incremental Hydrologic Inflows on Subbasin category, Cumulative Hydrologic Inflow is used to forecast and set the Forecasted Cumulative Hydrologic Inflow. If the Forecast Period method is not selected, the input Deterministic Incremental Hydrologic Inflow slot is used to forecast and set the Hydrologic Inflow Forecast series slot.

Slots Specific to this Method

☛ **Hydrologic Inflow Forecast**

Type: Series
Units: FLOW
Description: The forecasted hydrologic inflow values
Information:
I/O: Output only
Links: Not linkable

☛ **Deterministic Incremental Hydrologic Inflow**

Type: Series
Units: FLOW
Description: This slots holds a timeseries of the actual hydrologic inflows to the reach. These values are then adjusted by the forecast method and set on the Hydrologic Inflow Forecast slot.
Information: At the end of the run, the Hydrologic Inflow Forecast slot will be identical to this slot. If the Full Run method is selected for the Incremental Hydrologic Inflows on Subbasin category, values from the Incremental Hydrologic Inflow slot will be copied into this slot prior to the forecast. This slot is not used (i.e. inputs are ignored) if the Forecast Period method is selected.
I/O: Input or set to the values in the Incremental Hydrologic Inflow slot

Links: Not linkable

☛ Period of Perfect Knowledge

Type: Scalar
Units: FLOW
Description: Number of timesteps for which the forecast will equal the Deterministic Incremental Hydrologic Inflow, i.e., the forecast is known with complete certainty.
Information: Minimum value of 1; maximum value equal to the number of timesteps in the forecast period.
I/O: Input only
Type: Not linkable

☛ Recession Factor

Type: Scalar
Units: NONE
Description: A decimal value that is multiplied by the previous Hydrologic Inflow Forecast value to determine the current value after the Period of Perfect Knowledge.
Information:
I/O: Input only
Links: Not linkable

This method uses different slots and sets different slots if the Forecast Period is selected for the Incremental Hydrologic Inflows on Subbasin category. But, there is always a Source slot and a Target slot where the Source slot is input and used to forecasted and set the Target slot. If the Forecast Period is selected, the computational subbasin calls this method and uses the Cumulative Hydrologic Inflow (Source) to forecast and set the Forecasted Cumulative Hydrologic Inflow (Target). If it is not selected, the reservoir calls this method at the beginning of the timestep and uses the Deterministic Incremental Hydrologic Inflow (Source) slot to forecast and set the Hydrologic Inflow Forecast slot (Target). The following uses the Source/Target terminology to describe the methods.

The Source slot values are required inputs for each timestep. At the beginning of each controller timestep, the Geometric Recession method is executed. For each forecast timestep within the period of perfect knowledge, the Target is set to the Source value. For each forecast timestep after the period of perfect knowledge, the Target is set by multiplying the value of the Target from the previous timestep by the constant recession factor.

A value for the Source slot must be known at every timestep during the run. If Target values are desired past the end of the run, there must also be values in the Source slot at timesteps past the end of the run. If values for Source slot are not entered past the end of the run, the Target values for these timesteps are assumed to be zero.

17.1.16.3 Exponential Recession

On each timestep in the forecast period, this method will adjust the inflow hydrographs. If the Forecast Period method is selected for the Incremental Hydrologic Inflows on Subbasin category, the Cumulative Hydrologic Inflow is used to forecast and set the Forecasted Cumulative Hydrologic Inflow. If the Forecast Period method is not selected, the input Deterministic Incremental Hydrologic Inflow slot is used to forecast and set the Hydrologic Inflow Forecast series slot.

SLOTS SPECIFIC TO THIS METHOD

☛ HYDROLOGIC INFLOW FORECAST

Type: Series
Units: FLOW
Description: The forecasted inflow values computed from the Deterministic Incremental Hydrologic Inflow
Information:
I/O: Output only
Links: Not linkable

☛ DETERMINISTIC INCREMENTAL HYDROLOGIC INFLOW

Type: Series
Units: FLOW
Description: This slots holds a timeseries of the actual hydrologic inflows to the reach. These values are then adjusted by the forecast method and set on the Hydrologic Inflow Forecast slot.
Information: At the end of the run, the Hydrologic Inflow Forecast slot will be identical to this slot. If the Full Run method is selected for the Incremental Hydrologic Inflows on Subbasin category, values from the Incremental Hydrologic Inflow slot will be copied into this slot prior to the forecast. This slot is not used (i.e. inputs are ignored) if the Forecast Period method is selected.
I/O: Input or set to the values in the Incremental Hydrologic Inflow slot
Links: Not linkable

☛ FORECAST PERIOD

Type: Table
Units: NONE
Description: Number of timesteps, not including the current timestep, that the inflow hydrograph will be adjusted.
Information:
I/O: Input only
Links: Not linkable

☛ PERIOD OF PERFECT KNOWLEDGE

Type: Scalar
Units: FLOW
Description: Number of timesteps for which the forecast will equal the Deterministic Incremental Hydrologic Inflow, i.e., the forecast is known with complete certainty.
Information: Minimum value of 1; maximum value equal to the number of timesteps in the forecast period.
I/O: Input only
Links: Not linkable

☛ MINIMUM FORECASTED FLOW

Type: Series
Units: FLOW
Description: The minimum forecasted flow.
Information: If the computed value for Hydrologic Inflow Forecast is less than the Minimum Forecasted Flow, it is set to the Minimum Forecasted Flow.
I/O: Input only
Links: Not linkable

☛ LOW FLOW THRESHOLD

Type: Scalar
Units: FLOW
Description: The flow rate that dictates whether to use the Low Flow Recession Coefficient or the High Flow Recession Coefficient.
Information:
I/O: Input only
Links: Not linkable

☛ LOW FLOW RECESSION COEFFICIENT

Type: Scalar
Units: NONE
Description: The recession coefficient used when the Deterministic Incremental Hydrologic Inflow (at the end of the Period of Perfect Knowledge) is below or equal to the Low Flow Threshold.
Information:
I/O: Input only
Links: Not linkable

☛ HIGH FLOW RECESSION COEFFICIENT

Type: Scalar
Units: NONE

Description: The recession coefficient used when the Deterministic Incremental Hydrologic Inflow (at the end of the Period of Perfect Knowledge) is above the Low Flow Threshold.
Information:
I/O: Input only
Links: Not linkable

This method uses different slots and sets different slots if the Forecast Period is selected for the Incremental Hydrologic Inflows on Subbasin category. But, there is always a Source slot and a Target slot where the Source slot is input and used to forecasted and set the Target slot. If the Forecast Period method is selected, the computational subbasin calls this method and uses the Cumulative Hydrologic Inflow (Source) to forecast and set the Forecasted Cumulative Hydrologic Inflow (Target). If it is not selected, the reservoir calls this method at the beginning of the timestep and uses the Deterministic Incremental Hydrologic Inflow (Source) slot to forecast and set the Hydrologic Inflow Forecast slot (Target). The following uses the Source/Target terminology to describe the methods.

The Source slot values are input for each timestep. At the beginning of each controller timestep, the Exponential Recession method is executed. For each forecast timestep within the period of perfect knowledge, the Target is set to the Source value. For each forecast timestep after the period of perfect knowledge, the Target slot is set as described below:

$$ForecastedFlow = MAX\left[MinimumForecastedFlow, \left(Source \bullet e^{\frac{(-C)t}{T}}\right)\right]$$

where Source is the value in the Source slot at the end of the period of perfect knowledge, C is the recession coefficient, t is the elapsed time of the forecast period, and T is the total time from the end of the period of perfect knowledge to the end of the forecast period.

If the Source at the end of the period of perfect knowledge is negative, the Target at that timestep is exactly equal to the Source. However, the Source used in the recession equation, is the last positive value for Source. In the event that there is not a positive value for the Source, RiverWare issues a warning, and all values for Hydrologic Inflow Forecast within the forecast period will be set to the Minimum Forecasted Flow.

A value for the Source slot must be known at every timestep during the run. If the Target values are desired past the end of the run, there must also be values in the Source slot at timesteps past the end of the run. If values for Source are not entered past the end of the run, the Target values for these timesteps are assumed to be zero.

17.1.16.4 Coefficient and Exponent

On each timestep in the forecast period, this method will adjust the inflow hydrographs. If the Forecast Period method is selected for the Incremental Hydrologic Inflows on Subbasin category, the Cumulative Hydrologic Inflow is used to forecast and set the Forecasted Cumulative Hydrologic Inflow. If the Forecast Period methods is not selected, the input

Deterministic Incremental Hydrologic Inflow slot is used to forecast and set the Hydrologic Inflow series slot.

Slots Specific to this Method

☛ Hydrologic Inflow

Type: Series
Units: FLOW
Description: flow into the reservoir that is not gaged and/or does not enter through the main channel.
Information:
I/O: Output only
Links: Not linkable

☛ Hydrologic Inflow Adjust

Type: Series
Units: FLOW
Description: optional adjustment that can be made to the calculated Hydrologic Inflow
Information:
I/O: Optional; the Hydrologic Inflow Adjust may either be input by the user or it is set to zero if it is not input.
Links: Not linkable

☛ Hydrologic Inflow Net

Type: Series
Units: FLOW
Description: Sum of Hydrologic Inflow and Hydrologic Inflow Adjust
Information:
I/O: Output only
Links: Not linkable

☛ Forecast Inflow Parameters

Type: Table
Units: NONE
Description: Table slot that contains four parameters used in the forecast inflow method. The first row contains the values for the increasing hydrograph, the second row contains values for the decreasing hydrograph. The first column contains coefficients, the second column contains exponents.
Information: 2X2 table
I/O: Input only
Links: not linkable

☛ Deterministic Incremental Hydrologic Inflow

Type: Series
Units: FLOW
Description: This slots holds a timeseries of the actual Hydrologic Inflows to the reach. These values are then adjusted by the forecast method and set on the Hydrologic Inflow slot.
Information: At the end of the run, the Hydrologic Inflow Forecast slot will be identical to this slot. If the Full Run method is selected for the Incremental Hydrologic Inflows on Subbasin category, values from the Incremental Hydrologic Inflow slot will be copied into this slot prior to the forecast. This slot is not used (i.e. inputs are ignored) if the Forecast Period method is selected. The logic below uses the **Lower Bound** on the Deterministic Incremental Hydrologic Inflow slot as a minimum value. This is specified slot configuration (View->Configure menu). Consider setting this value as needed.
I/O: Input or set to the values in the Incremental Hydrologic Inflow slot
Links: not linkable

☛ Forecast Period

Type: Table
Units: None
Description: Number of timesteps, not including the current timestep, that the inflow hydrograph will be adjusted.
Information:
I/O: Input only
Links: not linkable

This method uses different slots and sets different slots if the Forecast Period method is selected for the Incremental Hydrologic Inflows on Subbasin category. But, there is always a Source slot and a Target slot where the Source slot is input and used to forecasted and set the Target slot. If Forecast Period method is NOT selected, the reservoir calls this method at the beginning of the timestep and uses the Deterministic Incremental Hydrologic Inflow (Source) slot to forecast and set the Hydrologic Inflow slot (Target). If it is selected, the computational subbasin calls this method and uses the Cumulative Hydrologic Infow (Source) to forecast and set the Forecasted Cumulative Hydrologic Inflow (Target). The subbasin then computes the incremental flow and sets the value on the Hydrologic Inflow slot. The following description uses the Source/Target terminology to describe the method.

The method works as follows: on the current timestep, the Target is set equal to the Source. The method then loops through the remaining timesteps in the forecast period and sets the Target using the following formula starting at i = 1:

$$HI_i = HI_{i-1} + HI_{i-1}\frac{(KI_i - KI_{i-1})}{KI_{i-1}}((C^i)^E)$$

where KI_i is the Source at timestep i, HI_i is the Target at timestep i. The counter i represents the timestep beyond the current timestep. For example, i = 1 is the next timestep, i = 2 is the current timestep + 2 timesteps, etc. The coefficient, C, and exponent, E, are the values in the Forecast Inflow Parameters slot. If(Source(i-1) <= Source(i)), E and C are the increasing (rising) values. Otherwise, E and C are the decreasing (falling) values.

In the above formula, there is a mathematical problem if KI_{i-1} is zero. In this situation, the Target at that index is set to the known inflow at that index. This allows the simulation to continue with reasonable values for the Target. The logic uses the Lower Bound on the Target slot as the minimum value. This is specified on the slot configuration (View->Configure menu). If this value is specified it is checked, otherwise only 0.0 is used in the check. The logic is:

If (Source(i-1) = 0.0 OR ABS(Source(i-1)) < ABS(Lower Bound)))

Target(i) = Source(i)

At the end of the method, the Hydrologic Inflow Net is set equal to the Hydrologic Inflow plus the Hydrologic Inflow Adjust for each timestep in the loop. This allows the rules to be able to use the Hydrologic Inflow Net before the object dispatches (such as in the GetMaxOutflowGivenInflow() function).

A value for the Source slot must be known at every timestep during the run. If forecasted Hydrologic Inflow values are desired past the end of the run, there must also be values in the Source slot at timesteps past the end of the run. If values for Source are not entered past the end of the run, the Hydrologic Inflows for these timesteps are assumed to be zero.

17.1.17 Incremental Hydrologic Inflows on Subbasin

The Incremental Hydrologic Inflows on Subbasin category contains methods used to specify that the reservoir has cumulative inflows that must be disaggregated into incremental inflows. There are two methods: Full Run and Forecast Period. The disaggregation is actually executed from the computational subbasin containing the reservoir. For more inforamation, click HERE (Section 7.1.22).

When the Input Hydrologic Inflow method is selected in the Hydrologic Inflow category, the Full Run becomes available. The method contains two slots: Cumulative Hydrologic Inflow and Incremental Hydrologic Inflow for the reservoir. Data must be input into the Cumulative Hydrologic Inflow slots. The computational subbasin will execute the Compute Full Run Incremental Hydrologic Inflows method and set the Incremental Hydrologic Inflow slots as input. Setting the Incremental Hydrologic Inflow slots as input prevents the slot values from being cleared in future model runs when the subbasin is disabled. The computational subbasin will first check that Full Run method has been selected on all reservoirs in the basin. During the model run, when the reservoir dispatches, the Full Run method will copy the value in the Incremental Hydrologic Inflow slot and set the Hydrologic Inflow slot.

When a forecast method is selected in the Generate Forecast Hydrology methods on the reservoirs, a new method will be available within the Incremental Hydrologic Inflows on Subbasin category: Forecast Period. This method will contain the Cumulative Hydrologic Inflow slot. When this method is selected the computational subbasin will execute the Compute Forecast Period Incremental Hydrologic Inflows method and set the Hydrologic Inflow slots.

17.1.17.1 None

This method is the default for the Incremental Hydrologic Inflows on Subbasin category and should be selected when hydrologic inflow data is not cumulative or the computation of incremental hydrologic inflows is not desired. There are no slots specifically associated with this method.

17.1.17.2 Full Run

The Full Run method is available from the Incremental Hydrologic Inflows on Subbasin method category on the reservoirs (storage reservoir, level power reservoir, and sloped power reservoir). This method is only available if the Input Hydrologic Inflow method is selected in the Hydrologic Inflow category. This method holds the slots necessary for the computation of incremental hydrologic inflows that is performed by the computational subbasin. The method contains two slots: Cumulative Hydrologic Inflow and Incremental Hydrologic Inflow. These slots will be accessed by the computational subbasin when executing the Compute Full Run Incremental Local Inflows method. The computational subbasin uses input Cumulative Hydrologic Inflow values to calculate and set the Incremental Hydrologic Inflow slot. Refer to the computational subbasin’s Incremental

Local Inflows documentation for details, click **HERE (Section 7.1.22.2)**. The Full Run method on the reservoir copies the results of the calculation in the Incremental Hydrologic Inflow slot values over to the Hydrologic Inflow series slot.

SLOTS SPECIFIC TO THIS METHOD

☛ CUMULATIVE HYDROLOGIC INFLOW

Type: Series
Units: FLOW
Description: The cumulative hydrologic inflow to the reservoir
Information: Hydrologic inflow is cumulative either (1) between headwater control points and the first reservoir in the river system and also throughout the system between two reservoirs, or (2) throughout the entire river system.
I/O: Required Input
Links: Not linkable

☛ INCREMENTAL HYDROLOGIC INFLOW

Type: Series
Units: FLOW
Description: The incremental hydrologic inflow to the reservoir
Information: This slot is set by the Calculate Incremental Flows method on the computational subbasin and represents the actual hydrologic inflow to the reservoir. If a Generate Forecast Hydrology method is selected, the Deterministic Incremental Hydrologic Inflow slot will be set to the values in this slot.
I/O: Computed and set with the Input flag
Links: Usually not linked

17.1.17.3 Forecast Period

The Forecast Period method is available from the Incremental Hydrologic Inflows on Subbasin category on the reservoir. This method is only available if the Forecast Hydrologic Inflows method is selected in the Hydrologic Inflow category and one of the forecasting methods (i.e. Geometric Recession, Exponential Recession, or Coefficient and Exponent) is selected in the Generate Forecast Hydrology category. If this method is selected, but the reservoir is not part of a subbasin with the appropriate methods selected, an error will be issued. The method contains two slots: Cumulative Hydrologic Inflow and Forecasted Cumulative Hydrologic Inflow. These slots will be accessed by the computational subbasin when executing the Compute Forecast Period Incremental Hydrologic Inflows method. The computational subbasin will use the user input Cumulative Hydrologic Inflow value to forecast and set the Forecasted Cumulative Hydrologic Inflow slot. It then uses this temporary value in its calculation of the incremental flows. The final result of this method (forecasted incremental hydrologic inflows) is set on the slot Hydrologic Inflow Forecast for each timestep in the forecast period. (Note: Hydrologic Inflow is set instead if the Coefficient and Exponent method is selected). Refer to the computational subbasin's Incremental Local Inflows documentation for details, click **HERE (Section 7.1.22.3)**.

SLOTS SPECIFIC TO THIS METHOD

☛ CUMULATIVE HYDROLOGIC INFLOW

Type: Series
Units: FLOW
Description: The cumulative hydrologic inflow to the reservoir
Information: Hydrologic inflow is cumulative either (1) between headwater control points and the first reservoir in the river system and also throughout the system between two reservoirs, or (2) throughout the entire river system.
I/O: Required Input
Links: Not linkable

☛ FORECASTED CUMULATIVE HYDROLOGIC INFLOW

Type: Series
Units: FLOW
Description: the forecasted cumulative Hydrologic inflow to the control point
Information: This slot is set by the selected Generate Forecast Hydrologic Inflows method on the control point as called from the computational subbasin. It represents the cumulative Hydrologic inflow to the control point forecasted throughout the forecast period. This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool **HERE (USACE_SWD.pdf, Section 5.5)**.
I/O: Output only
Links: Usually not linked

17.1.18 Evaporation and Precipitation

The Evaporation and Precipitation category methods are used to calculate the volume of Evaporation from and Precipitation to the surface of a reservoir over the timestep. Precipitation and Evaporation are used in the mass balance equations which are solved in the dispatch methods.

Some of the methods in this category only calculate evaporation.

17.1.18.1 None

The None method is the default method for the Evaporation and Precipitation category. It should be chosen if the user does not want to include Evaporation in the mass balance equation of the Reservoir. There are no slots specifically associated with this method. No calculations are performed by this method.

17.1.18.2 Daily Evaporation

The Daily Evaporation method is used to calculate the daily evaporation volume and the flow rate of the precipitation. The daily evaporation volume is a function of the Evaporation Rate, average Surface Area, and Pan Coefficient.

SLOTS SPECIFIC TO THIS METHOD

☛ **ELEVATION AREA TABLE**

Type: Table Slot
Units: LENGTH VS. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☛ **EVAPORATION**

Type: SeriesSlot
Units: VOLUME
Description: volume of water lost to evaporation during one timestep
Information:
I/O: Output only
Links: Not linkable

☛ **EVAPORATION TABLE**

Type: Table Slot
Units: NOUNITS VS. LENGTH (PER DAY)
Description: day of the year vs. Evaporation Rate

Information: The first of January is 0.
I/O: Required Input
Links: Not linkable

☛ Pan Evaporation Coefficient

Type: Table Slot
Units: NO UNITS
Description: a fractional value between 0 and 1 that represents the portion of potential evaporation which actually occurs
Information:
I/O: Required Input
Links: Not linkable

☛ Precipitation Rate

Type: SeriesSlot
Units: LENGTH vs. TIME
Description: precipitation intensity for the given timestep
Information:
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

☛ Precipitation Volume

Type: SeriesSlot
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☛ Surface Area

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area calculated from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

The Surface Area is determined using the Pool Elevation and the Elevation Area Table. The Evaporation Rate is looked up in the Evaporation Table according to the current day of the year. Evaporation is calculated using the following equation:

$$\text{Evaporation} = \text{Evaporation Rate} \times \text{Pan Evaporation Coefficient} \times (\text{Surface Area} + \text{Surface Area}(-1))/2$$

The volume of Precipitation that occurred over the timestep is then calculated with the following equation:

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)})/2$$

where in the above equations:

Evaporation Rate = the Evaporation Rate corresponding to the current day of the year

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

17.1.18.3 Input Evaporation

The Input Evaporation method should be used when the user wants to input the Evaporation Rate directly. This Evaporation Rate is used to compute the volume of water that evaporated over the timestep. Also, the user can input the evaporation volume directly on the Evaporation slot. In that case, the Evaporation Rate is not used to calculated Evaporation.

Slots Specific to this Method

☛ Elevation Area Table

Type: Table Slot
Units: LENGTH VS. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☛ Evaporation

Type: SeriesSlot
Units: VOLUME
Description: volume of water lost due to evaporation during the timestep
Information:
I/O: Output; optional input overrides calculation
Links: Not linkable

☛ Evaporation Rate

Type: SeriesSlot
Units: LENGTH PER TIME
Description: rate at which water evaporates from the surface
Information:

I/O: Optional input, disaggregated by method as described in the Evap and Precip Rate Specification category, or defaults to 0.0 if not specified by the user.
Links: Not linkable

☛ PRECIPITATION RATE
Type: SeriesSlot
Units: LENGTH PER TIME
Description: precipitation intensity for a given timestep
Information:
I/O: Optional input, disaggregated by method as described in the Evap and Precip Rate Specification category, or defaults to 0.0 if not specified by the user.
Links: Not linkable

☛ PRECIPITATION VOLUME
Type: SeriesSlot
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☛ SURFACE AREA
Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

At the beginning of the run, the chosen method in the **Evap and Precip Rate Specification** category is executed. This category allows you to specify the rates as monthly or periodic slots.

If the user specifies Evaporation directly (via input or rules), the value will be used instead of calculating a value below.

If Evaporation is not specified, the following equation is used to compute the volume of water that evaporated from the Reservoir over the timestep:

$$\text{Evaporation} = \text{Evaporation Rate} \times (\text{Surface Area} + \text{Surface Area }(-1))/2$$

The precipitation flow rate over the timestep is calculated as shown in the following equation:

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)}) / 2$$

where in the above equations:

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

17.1.18.4 Monthly Evaporation

In the Monthly Evaporation method, evaporation is calculated linearly from the Evaporation Coefficients entered for each month. This method will not work with a timestep longer than monthly. The total evaporated volume is a function of the average Reservoir Surface Area over the timestep, the Evaporation Coefficient, and the length of the timestep. The following slots are specifically associated with this method.

Slots Specific to this Method

☛ Elevation Area Table

Type: TableSlot
Units: LENGTH VS. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☛ Evaporation

Type: SeriesSlot
Units: VOLUME
Description: volume of water lost to evaporation during one timestep
Information:
I/O: Output only
Links: Not linkable

☛ Evaporation Coefficients

Type: TableSlot
Units: LENGTH PER TIME
Description: rate of evaporation for each month
Information: This slot contains one column of values. The Evaporation Coefficient for each month of the year must be input by the user beginning with the Evaporation Coefficient for January.
I/O: Required input
Links: Not linkable

☛ Precipitation Rate

Type: SeriesSlot
Units: LENGTH PER TIME
Description: precipitation intensity for the given timestep
Information: Value must be input by the user for each timestep.
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

☛ Precipitation Volume

Type: SeriesSlot
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☛ Surface Area

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area calculated from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

The Surface Area of the Reservoir is calculated based on the Elevation Area Table. The Evaporation is then calculated using the following formula:

$$\text{Evaporation} = \text{Evaporation Coefficient} \times (\text{Surface Area} + \text{Surface Area (-1)})/2 \times TimestepLength$$

The volume of Precipitation that occurred over the timestep is then calculated using the following equation:

$$\text{Precipitation} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)})/2 \times TimestepLength$$

where in the above equations:

Evaporation Coefficient = the Evaporation Coefficient for the current month

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

17.1.18.5 Pan and Ice Evaporation

The Pan and Ice Evaporation method is used to calculate the volume of evaporation with one of two methods based on the value of the Pan Ice Switch slot for each timestep. The Pan Ice Switch slot is used as an indicator of whether ice is present on the surface of the reservoir. A value of 1.0 in the Pan Ice Switch slot indicates that there is ice cover on the Reservoir that must be taken into account when Evaporation is calculated. A value of 0.0 or any number other than 1.0 in the Pan Ice Switch slot indicates that there is no ice on the surface of Reservoir. The following slots are those specifically associated with this method.

Slots Specific to this Method

☛ Evaporation

Type: Series
Units: VOLUME
Description: volume of water lost to evaporation during the current timestep
Information:
I/O: Output only
Links: Not linkable

☛ Elevation Area Table

Type: Table
Units: LENGTH vs. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☛ K Factor

Type: Series Slot with Periodic Input
Units: VELOCITY per TEMPERATURE_F
Description: factor relating average temperature, in degrees Fahrenheit, to evaporation rate
Information: This slot is a series slot, but the data can be input as a periodic relationship.
I/O: Optional but is required Input if the Pan Ice Switch slot is 1.0
Links: Not linkable

☛ Max Air Temperature

Type: Series
Units: TEMPERATURE in FARENHEIT
Description: maximum air temperature during the timestep
Information:
I/O: Optional; required only if the Pan Ice Switch slot is 1.0

Links: Not linkable

☛ MIN AIR TEMPERATURE
Type: Series
Units: TEMPERATURE IN FARENHEIT
Description: minimum air temperature during the timestep
Information:
I/O: Optional; required if the Pan Ice Switch slot is 1.0
Links: Not linkable

☛ PAN EVAPORATION
Type: Series
Units: LENGTH PER TIME
Description: evaporation rate from the surface
Information:
I/O: Optional; only required if the Pan Ice Switch is 0.0
Links: Not linkable

☛ PAN EVAPORATION COEFFICIENT
Type: Table
Units: DECIMAL
Description: weighing factor for pan evaporation rate
Information:
I/O: Optional; required if the Pan Ice Switch slot is 0.0
Links: Not linkable

☛ PAN ICE SWITCH
Type: Series
Units: NO UNITS
Description: indicator of surface ice coverage for each timestep; **1.0** = ice; any other number or **0.0** = no ice.
Information: This slot is a series slot, but the data can be input as a periodic relationship.
I/O: Required input
Links: Not linkable

☛ PRECIPITATION RATE
Type: Series
Units: LENGTH PER TIME
Description: precipitation intensity for a given timestep
Information:
I/O: Optional; defaults to 0.0 if not specified by the user.

Links: Not linkable

☛ Precipitation Volume

Type: Series
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☛ Surface Area

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

☛ Surface Ice Coverage

Type: SeriesSlot
Units: DECIMAL
Description: fraction of the Surface Area which is covered by ice
Information:
I/O: Optional; only used if the Pan Ice Switch slot is 1.0. Defaults to 0.0 for any timestep not specified by the user.
Links: Not linkable

If the Pan Ice Switch slot is equal to 1.0, ice is present and the following calculation is performed to compute evaporation:

$$\text{Evaporation} = \frac{\text{Max Air Temperature} + \text{Min Air Temperature}}{2} \times \text{K Factor} \times (1 - \text{Surface Ice Coverage}) \times \text{average Surface Area} \times \mathit{Timestep}$$

If the calculated Evaporation is less than zero, the Evaporation is set equal to zero.

The Precipitation is calculated with the following equation if the Pan Ice Switch slot is equal to 1.0:

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times (1 - \text{Surface Ice Coverage}) \times \text{average Surface Area}$$

The volume of precipitation that accumulated over the timestep at the Reservoir (Precipitation Volume) is the product of the precipitation flow rate and the timestep.

If the Pan Ice Switch slot is 0.0 or any number other than 1.0, there is no ice and the following calculation is performed to compute Evaporation:

$$\text{Evaporation} = \text{Pan Evaporation} \times \text{Pan Evaporation Coefficient} \times \text{average Surface Area} \times \text{Timestep}$$

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times \text{average Surface Area}$$

The volume of precipitation that accumulated over the timestep at the Reservoir (Precipitation Volume) is the product of the precipitation flow rate and the timestep.

17.1.18.6 Pan and Ice Evaporation, Current Surface Area

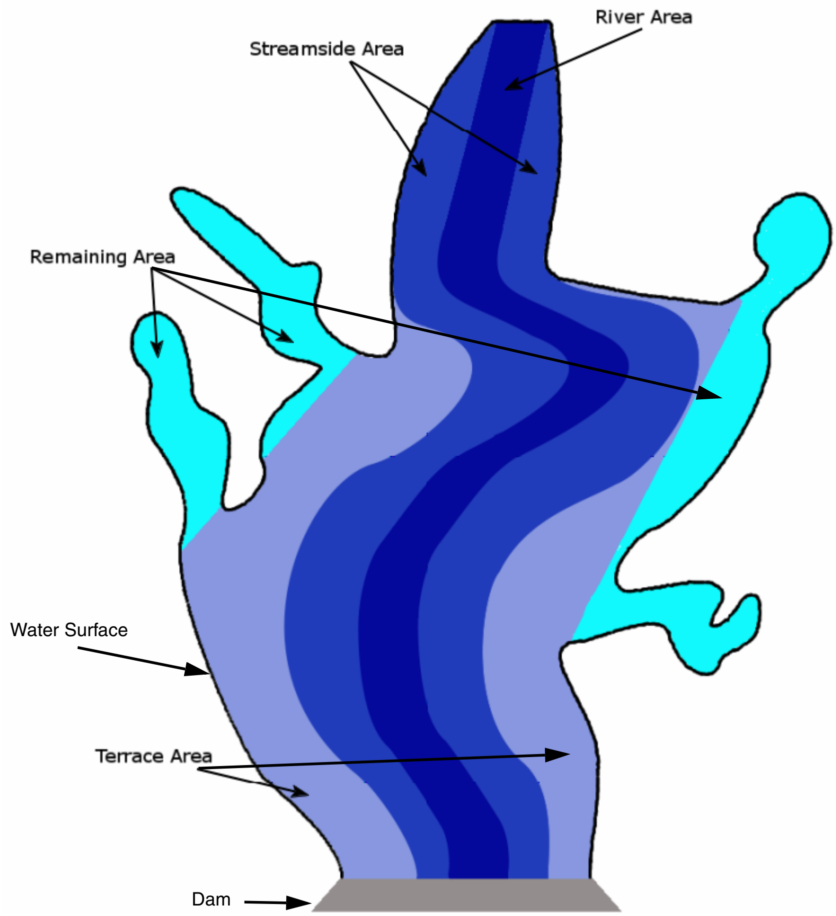
This method is exactly the same as the Pan and Ice Evaporation method. It uses the same slots, has the same required inputs and performs the same calculations. The only difference is that this method uses the instantaneous, end of timestep surface area instead of the average surface area over the timestep.

17.1.18.7 Periodic Net Evaporation

Periodic Net Evaporation computes the gross evaporation from the reservoir and then subtracts out components of evaporation that would have occurred if the reservoir had not been built. This is the net evaporation and is set in the **Evaporation** slot. Each area of the submerged reservoir is separate including:

- River
- Streamside
- Terrace, and any
- Remaining areas

Each can have a separate evaporation coefficient and possibly additional components in its computation like temperature. The area of each region is specified in a separate table relating reservoir pool elevation to each region's area. It is assumed that any precipitation that falls on the Remaining Area would have completely evaporated. The figure shows a sample of the different reservoir areas used in this method.



Slots Specific to this Method

☛ Average Precipitation

Type: Periodic
Units: Velocity (Length per Time)
Description: Slot describing the average precipitation
Information: Typically this would have a yearly period and monthly precipitation values. It is used in the computation of Remaining Evaporation.
I/O: Required Input
Links: Not Linkable

☛ Average Air Temperature

Type: Periodic
Units: TemperatureInFahren
Description: Slot describing the average air temperature

Information: Typically this would have a yearly period and monthly temperature values. It is used in computation of Streamside Evaporation and Terrace Evaporation.
I/O: Required Input
Links: Not Linkable

☛ Elevation Area Table
Type: Table
Units: Length vs Area
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required Input
Links: Not Linkable

☛ Elevation River Area
Type: Table
Units: Length vs Area
Description: Table relating reservoir Pool Elevation to submerged river area.
Information:
I/O: Required Input
Links: Not Linkable

☛ Elevation Streamside Area
Type: Table
Units: Length vs Area
Description: Table relating reservoir Pool Elevation to submerged streamside area.
Information:
I/O: Required Input
Links: Not Linkable

☛ Elevation Terrace Area
Type: Table
Units: Length vs Area
Description: Table relating Pool Elevation to submerged terrace area
Information:
I/O: Required Input
Links: Not Linkable

☛ Evaporation
Type: Series
Units: Volume

Description: Water lost from the reservoir to evaporation. This is the net evaporation and is the value that is included in the reservoir mass balance.
Information: This is the calculated as Gross Evaporation minus Salvage Evaporation
I/O: Output only
Links: Not Linkable

☛ Gross Evaporation
Type: Series
Units: Volume
Description: The total evaporation off the reservoir surface. This is the evaporation that is actually occurring from the reservoir.
Information: This is calculated as GrossEvaporationCoeff times SurfaceAreaAvg converted from a flow to volume.
I/O: Output only
Links: Not Linkable

☛ Gross Evaporation Coefficient
Type: Periodic
Units: Velocity (Length per Time)
Description: A table that describes the gross evaporation coefficient as it varies periodically. This is similar to a pan evaporation coefficient.
Information:
I/O: Required Input
Links: Not Linkable

☛ River Evaporation Coefficient
Type: Periodic
Units: Velocity (Length per Time)
Description: A table that describes the river evaporation coefficient as it varies periodically. This is similar to a pan evaporation coefficient.
Information:
I/O: Required Input
Links: Not Linkable

☛ Salvage Evaporation
Type: Series
Units: Volume
Description: The evaporation that would have occurred if the reservoir were not in place.

$$SalvageEvaporation = RiverEvaporation + StreamsideEvaporation + TerraceEvaporation + RemainingEvaporation$$

Information:
I/O: Output Only
Links: Not Linkable

☛ STREAMSIDE COEFFICIENT
Type: Periodic
Units: VELOCITYPERTEMPERATURE_F (I.E. LENGTH PER TIME PER TEMPERATURE_F)
Description: Periodic table of coefficients
Information: Typically this represents a unit depth per month per degree Fahrenheit (e.g. inches per month per degree Fahrenheit)
I/O: Required Input
Links: Not Linkable

☛ SURFACE AREA
Type: Series
Units: AREA
Description: Reservoir surface area computed from a lookup on the Elevation Area table
Information:
I/O: Output only
Links: Not Linkable

☛ TERRACE COEFFICIENT
Type: Periodic
Units: VELOCITYPERTEMPERATURE_F (I.E. LENGTH PER TIME PER TEMPERATURE_F)
Description: Periodic table of coefficients
Information: Typically this represents a unit depth per month per degree Fahrenheit (e.g. inches per month per degree Fahrenheit)
I/O: Required Input
Links: Not Linkable

The method will be passed in a current estimate of Surface Area and Average Surface Area. The latter is an average of the current estimate and previous timestep's value. In this description, it is called SurfaceAreaAvg. Similarly, StreamsideAreaAvg, RiverAreaAvg, TerraceAreaAvg and RemainingAreaAvg are all averages of the current and previous values.

In the following steps, the FlowToVolume and VolumeToFlow notation indicates that the specified expression will be converted from a flow to a volume (or vice versa) using the timestep length. This is necessary for the units to work correctly as evaporation [Volume units] is computed as a coefficient [Length/Time units] times an area [L^2 units]. Note that in the following steps, the slots are in bold while intermediate values are not.

The method does the following:

1. Get the value from the periodic **Gross Evaporation Coefficient** slot. If not valid, issue an error.
2. Compute **Gross Evaporation**:

$$GrossEvaporation = FlowToVolume(GrossEvaporationCoefficient \times SurfaceAreaAvg)$$

3. Get the value from the periodic **River Evaporation Coefficient** slot. If not valid, issue an error.
4. Look up the Pool Elevation at t and t-1 on the **Elevation River Area** table to get the RiverArea at t and t-1. Then

$$RiverAreaAvg = \frac{RiverArea[t] + RiverArea[t-1]}{2}$$

5. Compute River Evaporation:

$$RiverEvaporation = FlowToVolume(RiverEvaporationCoefficient \times RiverAreaAvg)$$

This simulates that the river evaporation that would have occurred without the reservoir is a function of area and coefficient.

6. Get the value from the periodic **Streamside Coefficient** slot. If not valid, issue an error.
7. Look up the Pool Elevation at t and t-1 on the **Elevation Streamside Area** table to get the StreamsideArea at t and t-1. Then:

$$StreamsideAreaAvg = \frac{StreamsideArea[t] + StreamsideArea[t-1]}{2}$$

8. Get the value from the periodic **AverageTemperature** slot. If not valid, issue an error.
9. Compute Streamside Evaporation:

$$StreamsideEvaporation = FlowToVolume(StreamsideCoefficient \times StreamsideAreaAvg \times AverageAirTemperature)$$

This simulates that the streamside evaporation that would have occurred without the reservoir is a function of area, coefficient, and average air temperature.

10. Get the value from the periodic **Terrace Coefficient** slot. If not valid, issue an error.

11. Look up the Pool Elevation at t and t-1 on the **Elevation Terrace Area** table to get the Terrace Area at t and t-1. Then:

$$TerraceAreaAvg = \frac{TerraceArea[t] + TerraceArea[t-1]}{2}$$

12. Compute Terrace Evaporation:

$$TerraceEvaporation = FlowToVolume(TerraceCoefficient \times TerraceAreaAvg \times AverageAirTemperature)$$

This simulates that the terrace evaporation that would have occurred without the reservoir is a function of area, coefficient, and average air temperature.

13. Compute the average Remaining Area as:

$$RemainingAreaAvg = SurfaceAreaAvg - RiverAreaAvg - StreamsideAreaAvg - TerraceAreaAvg$$

If RemainingAreaAvg is less than zero, an error will be issued as the table data is incorrect.

14. Get the value from the periodic **Average Precipitation**. If not valid, issue an error.

15. Compute RemainingEvaporation:

$$RemainingEvaporation = FlowToVolume(RemainingArea \times AveragePrecipitation)$$

This simulates that all of the precipitation on the Remaining Area would have evaporated.

16. Compute **Salvage Evaporation**:

$$SalvageEvaporation = RiverEvaporation + StreamsideEvaporation + TerraceEvaporation + RemainingEvaporation$$

17. Compute **Evaporation** as follows:

$$Evaporation = GrossEvaporation - SalvageEvaporation$$

The **Evaporation** is then a volume that is removed from the reservoir mass balance in the dispatch method.

17.1.18.8 Single Evaporation

In the Single Evaporation method, evaporation is calculated linearly from the Single Evaporation Coefficient entered by the user. The total evaporated volume is a function of the average Reservoir Surface Area over the timestep, the Single Evaporation Coefficient, and the length of the timestep. The following slots are specifically associated with this method.

Slots Specific to this Method

☛ Elevation Area Table

Type: Table Slot
Units: LENGTH VS. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☛ Evaporation

Type: SeriesSlot
Units: VOLUME
Description: volume of water lost to evaporation during one timestep
Information:
I/O: Output only
Links: Not linkable

☛ Single Evap Coeff

Type: Table Slot
Units: LENGTH PER TIME
Description: rate of evaporation
Information: This slot contains a single value that represents the evaporation rate.
I/O: Required input
Links: Not linkable

☛ Precipitation Rate

Type: SeriesSlot
Units: LENGTH PER TIME

Description: precipitation intensity for the given timestep
Information: Value must be input by the user for each timestep.
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

☛ PRECIPITATION VOLUME

Type: SeriesSlot
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☛ SURFACE AREA

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area calculated from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

The Surface Area of the Reservoir is calculated based on the Elevation Area Table. The Evaporation is then calculated using the following formula:

$$\text{Evaporation} = \text{Evaporation Coefficient} \times (\text{Surface Area} + \text{Surface Area (-1)})/2 \times TimestepLength$$

The volume of Precipitation that occurred over the timestep is then calculated using the following equation:

$$\text{Precipitation} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)})/2 \times TimestepLength$$

where in the above equations:

Evaporation Coefficient = SingleEvapCoeff entered by the user

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

17.1.19 Evap and Precip Rate Specification

This category allows you to choose how the evaporation and precipitation rates will be specified. The category is only available when the **Input Evaporation** method in the **Evaporation and Precipitation** category is specified.

17.1.19.1 None

This is the default method; that is, the rates must be input, set by a rule, or they default to 0.0.

17.1.19.2 Monthly Rates

This method allows you to specify the evaporation and precipitation rates as a series of monthly values for the entire run.

☛ **EVAPORATION RATE MONTHLY**

Type: Series Slot
Units: VELOCITY
Description: The evaporation rate for each month of the run.
Information: You must set the timestep for this series slot to be monthly. Because this slot is monthly, it is most likely different than the run timestep. As a result, if you “synchronize objects”, you must select the toggle in the synchronization control to “Exclude Slots with Different Timestep from Run.” This will prevent changing the timestep of this slot when other slots are synchronized.
I/O: Optional input
Links: Not linkable

☛ **PRECIPITATION RATE MONTHLY**

Type: Series Slot
Units: VELOCITY
Description: The precipitation rate for each month of the run.
Information: You must set the timestep for this series slot to be monthly. Because this slot is monthly, it is most likely different than the run timestep. As a result, if you “synchronize objects”, you must select the toggle in the synchronization control to “Exclude Slots with Different Timestep from Run.” This will prevent changing the timestep of this slot when other slots are synchronized.
I/O: Optional input
Links: Not linkable

METHOD DETAILS

At the beginning of run, the method disaggregates the **Evaporation Rate Monthly** and **Precipitation Rate Monthly** to the **Evaporation Rate** and **Precipitation Rate** slots, respectively. If the timestep of the run is monthly, it uses the values directly. If the timestep of the run is less than a month, it **look ups** the month that contains the given timestep and uses that value. No interpolation is performed.

If the run timestep is annual, an error is issued.

If the two slots are not monthly but have inputs, an error is issued.

If there is no value in the monthly slot for a given month, then the rate is set to 0.0.

17.1.19.3 Periodic Rates

This method allows you to specify the evaporation and precipitation rates as a periodic relationship.

☛ Evaporation Rate Periodic

Type: Periodic Slot
Units: VELOCITY
Description: The evaporation rate as a periodic relationship.
Information: Like other periodic slots, you can choose the period and whether to interpolate or lookup.
I/O: Required Input
Links: Not linkable

☛ Precipitation Rate Periodic

Type: Periodic Slot
Units: VELOCITY
Description: The precipitation rate as a periodic relationship.
Information: Like other periodic slots, you can choose the period and whether to interpolate or lookup. If you do not wish to model precipitation, you still must enter a zero in this periodic slot.
I/O: Required Input
Links: Not linkable

Method Details

At the beginning of run, the method sets the **Evaporation Rate** and **Precipitation Rate** slots by looking up (or interpolating as configured on the periodic slot) the given timestep in the **Evaporation Rate Periodic** and **Precipitation Rate Periodic** slots, respectively. If accessing the periodic slot fails due to missing values, then an error is issued and the run stops.

17.1.20 Low Flow Releases

This category is only used to add the slots necessary for low flow release calculations. These slots are generally used by a RPL function (called MeetLowFlowRequirement)to compute the low flow releases necessary to meet the low flow requirements on control point objects.

17.1.20.1 None

This method performs no calculations and adds no slots.

17.1.20.2 Enable Low Flow Releases

This method performs no calculations. It simply adds the Low Flow Release slot and Maximum Low Flow Delivery Rate slot. Use of this method for USACE-SWD is described HERE (USACE_SWD.pdf, Section 3.7).

Slots Specific To This Method

☛ Low Flow Release

Type: Series Slot
Units: FLOW
Description: The portion of the Outflow that is intended to meet a low flow requirement
Information: This slot is normally computed by a RPL function (MeetLowFlowRequirement) that computes the low flow releases necessary to meet the low flow requirements on control point objects.
I/O: Usually set by a rule
Links: Not linkable

☛ Maximum Low Flow Delivery Rate

Type: Periodic Slot
Units: FLOW
Description: The maximum low flow delivery rate for the reservoir
Information: This value is used by the RPL function (MeetLowFlowRequirement) that determines the low flow releases from each reservoir. Low flow releases will be limited to this value.
I/O: Required input
Links: Not linkable

17.1.21 Surcharge Release

Surcharge releases methods determine releases made during the forecast period. The releases are considered mandatory, due to the volume of water in the reservoir. These methods are executed only from the solveMB_givenInflowOutflow dispatch method when the Outflow slot is set with the surcharge release (S) flag. This flag can only be set by a rule.

17.1.21.1 None

The default. Nothing is done

17.1.21.2 Flat Top Surcharge

This method follows the procedure used by the computer program SUPER and the US Army Corps of Engineers Southwest District as described **HERE (USACE_SWD.pdf, Section 3.3)**.

SLOTS SPECIFIC TO THIS METHOD

☛ **OPERATING LEVEL TABLE**

Type: Periodic
Units: TIME VS LENGTH AT OPERATING LEVELS
Description: table of describing the seasonal variation of elevation (storage) in a reservoir at each of the user-designated operating levels.
Information: number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep. See **HERE (Section 16.1.23)** for a method to modify which operating level table is used within a run.
I/O: Required Input
Links: Not Linkable

☛ **FORECAST PERIOD**

Type: Scalar
Units: NO UNITS
Description: number of timesteps in the forecast period.
Information: number of timesteps over which Inflows are forecast and flood control releases (including surcharge releases) are calculated; includes current timestep.
I/O: Required Input

Links: Not Linkable

☛ Top of Conservation Pool

Type: Scalar
Units: No Units
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the conservation pool.
Information: COE-SWD has used 5 (default) in the past, but can be any number.
I/O: Required Input
Links: Not Linkable

☛ Rating Curves

Type: Table
Units: Storage vs Flow (Induced Surcharge) vs Flow (Free-Flow)
Description: two curves representing the induced surcharge (minimum) and free-flow (maximum) that can be released based on elevation in the reservoir. The intersection of the two curves must be defined (there must be a row where the values in the middle and third column are equal) and the points of the free-flow curve beyond the end of the induced surcharge curve must be input in the table column for the induced surcharge curve.
I/O: Required input unless the method HERE (Section 17.1.22.2) is selected.
Links: Not Linkable

☛ Surcharge Release

Type: Series
Units: Flow
Description: release required by operating when the reservoir elevation is in the surcharge pool.
Information: set by surcharge release method for all timesteps in forecast period
I/O: Output
Links: Not Linkable

☛ Minimum Mandatory Release

Type: Series
Units: Flow
Description: minimum surcharge release that can be achieved within the timestep following the induced surcharge curve
Information: calculated for each timestep in the forecast period, but the slot is set only for the current timestep.
I/O: Output only
Links: Not linkable

☛ Maximum Mandatory Release

Type: Series
Units: Flow
Description: maximum surcharge release that can be achieved within the timestep following the free-flow rating curve
Information: calculated for each timestep in the forecast period, but the slot is set only for the current timestep.
I/O: Output only
Links: Not linkable

This method is based on the methodology developed by the US Army Corps of Engineers, Southwest Division. In this method, the surcharge, or mandatory, releases, as well as minimum and maximum mandatory releases, are determined for a number of timesteps called the forecast period that will prevent overtopping of a reservoir. These releases are determined for each timestep in the Run Control Dialog.

The name of this method comes from the process of selecting releases during the forecast period to minimize future releases. In other words, release a little more today to reduce, or flatten out, future releases. This approach aims to prevent excessively large inflows from creating equally large releases by extending the time period over which these inflows are released.

This method is used by the dispatch method, solveMB_givenInflowOutflow, whenever the surcharge release flag (S) is set on the Outflow slot (this flag can only be set by a rule). When inflows, which includes the sum of upstream inflows and hydrologic inflows, are known, and the rule sets the flag for the current controller timestep, the surcharge method is launched and surcharge releases and minimum and maximum mandatory releases are calculated for each timestep in the forecast period for the current controller timestep. This process progresses through each timestep on the Run Control Dialog.

The user should be aware of the distinction between the controller timestep, or timesteps set in the Run Control Dialog, and forecast timesteps. Once the surcharge method is invoked, surcharge calculations are performed for the forecast timesteps. The controller timestep does not advance, even though the Surcharge Release slot receives output from the method. Outputs in this slot are overwritten during the next controller timestep, except for the first.

The calculation of the surcharge releases for the forecast period is accomplished by using a loop structure once the surcharge method is invoked by the dispatch method mentioned above. Calculations for each timestep in the forecast period are exactly the same except for changes in the starting storage volume and inflows. The following sections provide a detailed description of the process that is used to calculate the surcharge release and minimum and maximum mandatory releases within.each loop. There a three main sections: calculations of the minimum and maximum mandatory releases, which bound the surcharge release, and the calculation of the surcharge release.

The first step is to determine the “forecast storage”. This volume is the initial storage (storage at end of previous computational timestep if this is the first time through the loop, or the storage at the end of the previous forecast timestep for all subsequent loops) plus the total inflow minus any surcharge releases for all forecast timesteps prior to the current forecast timestep. The total inflow is the sum of the upstream inflows plus hydrologic inflows for the current forecast period. For the first forecast timestep, the forecast storage is simply the storage in the reservoir on the previous controller timestep. On subsequent forecast timesteps, the forecast storage includes total inflows through the current forecast timestep and surcharge releases through the previous forecast timestep.

A table interpolation of the Elevation Volume Table is then done to determine the forecast elevation. If the forecast elevation is less than the lowest capacity value input in the induced surcharge curve, which is the middle column on the Rating Curves slot, the minimum mandatory release is zero and the program progresses to determine the maximum mandatory release. Otherwise it continues to determine the minimum mandatory release.

Calculation of the minimum mandatory release

If the forecast elevation falls between the highest and lowest capacity (or elevation) values on the induced surcharge curve, the program determines a discharge rate corresponding to the forecast elevation by interpolation of capacity values from the induced surcharge curve. This is the minimum that can be released based on the forecast elevation. If the forecast elevation is greater than the highest capacity value, the curve must be extended, i.e., there is no extrapolation of the curve. In a similar fashion, the total inflow is used in an interpolation of the induced surcharge curve to determine a corresponding elevation. If this elevation corresponding to the total inflow is less than the minimum discharge rate based on the forecast elevation, the pool is falling. For the opposite case, the pool is rising. If the total inflow and the minimum release corresponding to the forecast elevation are equal, the minimum release is set to this discharge rate and the program goes on to determine the maximum release.

Falling pool: If the pool is falling (the forecast inflow is less than minimum discharge corresponding to the forecast elevation), the program uses a specific portion of the induced surcharge curve to determine the minimum mandatory release. The set of points includes all points between and including the point on the curve corresponding to the total inflow for the current forecast timestep to the minimum discharge based on the forecast elevation. Using this portion of the induced surcharge curve, the program calculates how long it will take the reservoir to move from the forecast elevation to an elevation corresponding to the total inflow or how much volume can be discharged within one timestep, whichever is less. In the case of a falling pool, the process involves moving down the curve. For the opposite case (rising pool), the process will be reversed, moving “up” the curve since the discharge rate corresponding to the forecast elevation is less that the total inflow. In either case, the program begins at the curve point corresponding to the forecast storage and progresses to that point corresponding to the total inflow.

The process of progressing through the curve point is as follows:

First, the average of the discharge rates corresponding to the first two (or highest) curve points is found. Then the time it would take to release a certain volume at some discharge rate is determined. This volume is the difference in storage values from the curve points that correspond to the two discharges used to calculate the average discharge rate. The time is calculated by the following formula:

$$Time = \frac{Storage(i) - Storage(i-1)}{(TotalInflow - AverageDischarge)}$$

where

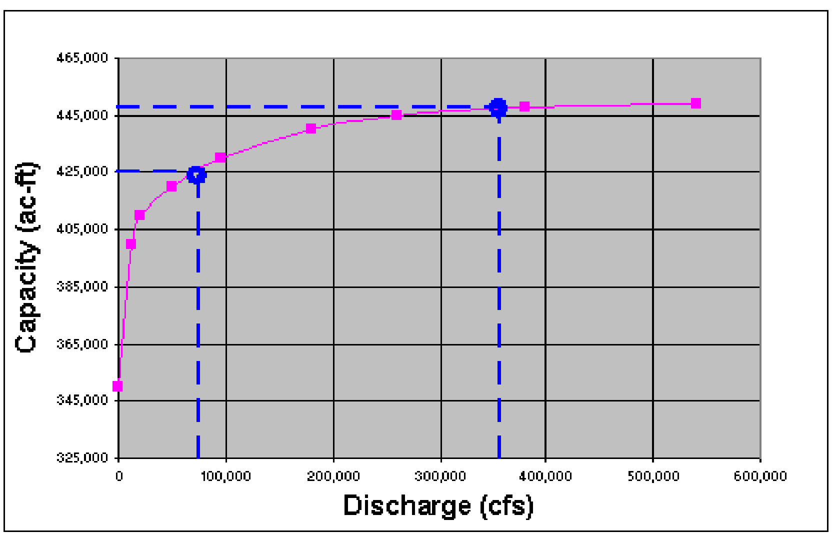
Storage(i) and Storage(i-1) are the storage values corresponding to the two discharge values (the difference is the volume that is being released). The value of Forecast Inflow - Average Discharge represents the net discharge.

If this time is less than one timestep's time, the program stores the time and the total volume that could be released in this time at the average discharge, which equal Time * Average Discharge. The program then moves to the next two curve points, gets another average discharge rate corresponding to these two curve points and again determines the time it would take to drop from one capacity value to the next at the net discharge rate. This new time plus the time for the previous two curve points is now the new total time and is again checked to see if it is greater than one timestep. If less still less than one timestep, the total time and total volume are stored (replacing the previous values) and the program moves to the next two curve points. This process continues until either the point corresponding to the total inflow is reached or the total time has reached or exceeded one timestep, whichever occurs first.

If the total time at this point is exactly equal to one timestep, the calculations are done and the minimum mandatory release equals the total volume divided by the time of one timestep. If the total time is greater than one timestep's time, the total time from the previous iteration is used to see how much time was left before reaching a total time of one timestep. Given the average discharge rate for the current iteration, the volume that could be released within the time remaining is determined. This volume is then added to the total volume and the final minimum mandatory discharge rate is calculated as described above.

If the computation gets through all the curve points, i.e., the curve point corresponding to the total inflow, with a total time that is less than or equal to one timestep, the total volume is increased by adding to the previously stored total volume to (Time of 1 timestep - total time)*total inflow. Then, the minimum mandatory release is calculated by dividing the total volume by the time of one timestep.

Example of minimum mandatory release w/ falling pool: Below is a plot of the induced surcharge curve. Assume that the timestep is a day, there is a falling pool, the total inflow for the current forecast timestep = 80,000 cfs, and the minimum release that the reservoir can make corresponding to the forecast storage of 450,000 ac-ft. = 365,000 cfs. Therefore, the portion of the curve that is used to determine the minimum mandatory release is bracketed by these two discharge points (donuts on plot), and the three actual curve points between these. First, the average discharge rate of 365,000 and 260,000 cfs = 312,500 cfs is determined. Then the volume between the two storage values corresponding to these two discharge points (450,0000 - 445,000 = 5,000 ac-ft = 217,800,000 cu. ft.) is calculated. Then, the time it takes the reservoir to release this volume of water at the calculated average discharge rate (217,800,000cu.ft./312,500 cfs = 0.26 hours) is determined. Since this time is less than 24 hours (one timestep), the total time of 0.26 hours and the total volume = 0.26*312,500 = 5,000 ac-ft is stored and the next two discharge points are used. Using these two points (260,000 and 180,000), the average discharge rate = 220,000 cfs, the volume between storage values corresponding to these two discharges = 445,000 - 440,000 = 5,000 ac-ft, and the time to release that volume at the net discharge rate = 0.43 hours. This total time = 0.69 is still less than 24 hours, so the program moves to the next two points (180,000 and 95,000) after storing the total volume, which = 10,000 ac-ft. The average discharge for the next two points is 137,500 cfs and the storage volume between these points is 440,000 - 430,000 = 10,000 ac-ft. The time to release this volume at the net discharge rate is 2.10 hours, so the total time is now 2.79 hours, which is still less than 24 hours. The total volume that is stored now = 20,000 ac-ft). The final two discharge points are 95,000 and 80,000 cfs, with an average of 87,500 cfs. The volume between these points is 5,000 ac-ft (430,000 - 425,000 ac-ft). The time to release this volume is 8.07 hours, the total time is 10.86 hours, and the total volume = 25,000 ac-ft). Although still less than 24 hours, all of the specific points of the curve have been used. The final total volume is the sum of the previous total of 25,000 ac-ft + (24hrs - 10.86)*total inflow. The minimum mandatory release is this new total volume divided by 24 hours.



Rising Pool: If the pool is rising (total inflow greater than discharge corresponding to the forecast storage), the calculation is the same as for the case of a falling pool except that the progression through the curve point will be up the curve instead of down the curve. The lowest point is the minimum discharge that must be released corresponding to the forecast storage and the highest point is the total inflow for the current forecast day.

Calculation of the maximum mandatory release

The maximum release is determined in the same manner as the minimum release except that the free-flow rating curve is used instead of the induced surcharge curve. The program again determines if the pool is be rising or falling, then progressed through the specific curve points to calculate the maximum mandatory release.

Calculate surcharge releases

Once the maximum and minimum mandatory releases have been determined, the program computes surcharge release to “flat top” the incoming flood. The objective is to have slightly higher mandatory releases sooner than later to try to reduce the surcharge releases that would need to be made later as inflows continue to come into the reservoir. The program looks ahead in the forecast period to anticipate high flows with higher releases in order to reduce the maximum release needed in the future to keep the pool at or below the induced surcharge curve until such time as the top of the induced surcharge curve is reached.

Bracket mandatory release on induced surcharge curve: To determine the mandatory release, the program steps through the points on the induced surcharge curve to determine what the surcharge release must be in order to reduce the maximum release at some time in the future when the reservoir is expected to be the fullest.

First, the available storage is calculated as the difference between the forecast storage and the storage at the first induced surcharge point. Then, the highest storage in the reservoir over the remainder of the forecast period is calculated using the remaining inflow hydrograph and a constant release equal to the discharge rate corresponding to the currently considered point on the induced surcharge curve. If the highest storage is greater than the storage available, the program continues to the next point on the induced surcharge curve and recalculates the available storage and the highest forecast storage; again using the remaining inflow hydrograph and the release associated with that point. Moving to each new curve point effectively increases the release, there by reducing the maximum storage required and increasing the available storage. This process of checking each of the points on the induced surcharge curve continues until the highest forecast storage is less than the available storage.

Before moving to another point on the curve, the program stores the available storage, the highest forecast storage, and the discharge rate to be used for future interpolation.

If the final, or highest, curve point is reached and the available storage is still less than the highest forecast storage, the program execution is terminated and a request is made for extending the curve.

Once a curve point is found that satisfies the condition that the highest forecast storage is less than the available storage, the program is considered to have found the general vicinity of the appropriate value of the surcharge release, or in other words bracketed the portion of the curve. At this time, there are two pairs of stored values for each of the following parameters: available storage, the highest forecast storage, and the discharge rate.

Use forecast inflows to zoom in on solution: Having bracketed the part of the induced surcharge curve that contains the appropriate surcharge release, it is now necessary to find the point in that range for which the available storage is just equal to the highest forecast storage. This part of the program uses inflow values from the forecasted inflow array as intermediate discharge test points. The purpose of looking through the forecasted inflows narrows the solution for the right surcharge release.

The program scans through the forecast period to see if any inflows fall between the two discharge values that have been stored. If any fall in there, the smallest is stored. Then, this inflow value is used similarly to how the discharge values from the various curve points were used above to determine the available storage and highest forecast storage if the inflow is assumed to be the release. This is followed by a check to see if the available storage is less than the highest forecast storage. If so, the program looks to see if another higher inflow within the forecast period can satisfy the condition and, if found, available storage and highest forecast storage is recalculated. However, at the next point, the lower limit within which the inflow must fit above is raised to the inflow value that was used previously. If no inflow is large enough to make the available storage larger than the required storage, continue on with the two discharge values from the curve that were stored after the solution was initially bracketed.

If there are no inflows within the forecast period that fall in the range of the two discharge points on the curve or if no inflow values during the forecast period are large enough to make the available storage large enough to exceed the highest forecast storage, the algorithms goes to the next section where an interpolation is used to find the surcharge release for the current forecast day using points off the curve.

Interpolate to find surcharge release for current forecast day: The induced surcharge curve has been bracketed by two points. At the higher point on the curve, the discharge satisfies the condition that the available storage >= highest forecast storage. At the lower point, the condition is not satisfied. (The discharge values may be specified input points from the induced surcharge curve, or may be points found as described in the previous section from the forecasted inflow array.) Once the condition of available space being larger than or equal to the required storage is satisfied, the extra available storage is calculated for the set of points that satisfied the condition (available storage - highest forecast storage) and the extra volume needed to store the highest forecast storage for the set of points that didn't satisfy the condition (highest forecast storage - available storage). Finally, using the extra space, extra storage needed, and the two discharge points, an interpolation is done to find the surcharge release for the current forecast day. The resulting surcharge release value is that which would very closely match the available storage and highest forecast storage, given that release rate.

Compare surcharge release with maximum and minimum release values: Once the surcharge release is determined, it is compared to the maximum and minimum mandatory releases calculated earlier. If the surcharge release is greater than or less than these limits, respectively, it is set to the limit. In addition, the storage of the reservoir is reduced to reflect the surcharge release for the current forecast timestep. This storage for the current forecast timestep is used as the initial storage for the calculation of forecast storage on the next timestep in the forecast period.

Check surcharge release against conservation pool: The final section in this method involves a check to see of the surcharge release results in a pool level below top of conservation pool. The reservoir storage with the surcharge release already made through current forecast day is checked to see if it is above or below the conservation pool. If below, the surcharge release is set to that which would bring the storage down to the conservation pool. If the storage is already below the conservation pool before this release is made, the surcharge release for the current forecast timestep is zero. If above the conservation pool, the surcharge release is left as is.

The program then moves to the next forecast day. After the surcharge releases and minimum and maximum mandatory releases have been determined for every timestep in the forecast period, the calculations are complete.

Finally, the method selected in the Elevation Max Duration Constraints category HERE (Section 17.1.23) is executed. This may increase the Surcharge Release to avoid the reservoir from exceeding a desired elevation for more than the desired number of timesteps.

17.1.21.3 Induced Surcharge Curve

This method follows the procedure used by the US Army Corps of Engineers Kansas City office to determine surcharge releases during a forecast period for reservoirs with tainter gates.

Slots Specific to this Method

☛ Operating Level Table

Type: Periodic
Units: Time vs Length at Operating Levels
Description: table of describing the seasonal variation of elevation (storage) in a reservoir at each of the user-designated operating levels.
Information: number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep. See HERE (Section 16.1.23) for a method to modify which operating level table is used within a run.
I/O: Required Input
Links: Not Linkable

☛ Operating Level

Type: Series
Units: None
Description: The computed operating level
Information: This slot is computed using the pool elevation and the Operating Level Table
I/O: Output only
Links: Not Linkable

☛ Forecast period

Type: Scalar
Units: No Units
Description: number of timesteps in the forecast period.
Information: number of timesteps over which Inflows are forecast and flood control releases (including surcharge releases) are calculated; includes current timestep.
I/O: Required Input
Links: Not Linkable

☛ Top of Flood Pool

Type: Scalar
Units: No Units
Description: Operating level (as defined in Operating Level Table) corresponding to the top of flood pool.
I/O: Required Input
Links: Not Linkable

☛ Top of Conservation Pool

Type: Scalar
Units: No Units
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the conservation pool.
I/O: Required Input
Links: Not Linkable

☛ Induced Surcharge Curve

Type: Table
Units: Flow (Inflow) vs Length (Elevation) vs Flow (Surcharge Release)
Description: family of curves that relate elevation and surcharge release for various inflow values.
I/O: Required Input
Links: Not Linkable

☛ SURCHARGE RELEASE

Type: Series
Units: FLOW
Description: release required by operating when the reservoir elevation is in the surcharge pool.
Information: set by surcharge release method for all timesteps in forecast period
I/O: Output
Links: Not Linkable

☛ RECESSION TARGET

Type: Series
Units: LENGTH
Description: pool elevation at which surcharge operations are suspended.
Information: once pool drops to this elevation, Surcharge Release operations will terminate.
I/O: Optional, can be input, or will default to the elevation of the bottom of surcharge pool (top of flood pool).
Links: Not Linkable

☛ GATE OPENING CURVE

Type: Table
Units: LENGTH (ELEVATION) VS FLOW VS LENGTH(GATE OPENING).
Description: This is a family of curves of that relate pool elevation and discharge for various gate openings. This must be specified as a 3D table as described HERE (Three-dimensional data format).
I/O: Required Input
Links: Not Linkable

☛ GATE OPENING

Type: Series
Units: NO UNITS
Description: gate opening calculated for each timestep.
I/O: Output
Links: Not Linkable

☛ STORAGE TOLERANCE

Type: Scalar
Units: VOLUME
Description: allowable difference between calculated average storage and beginning of timestep storage.
Information:
I/O: Required Input

Links: Not Linkable

This method determines a surcharge release for each timestep in the forecast period. The process begins by determining a Surcharge Release value from the Induced Surcharge Curve, given Inflow (net) and current Pool Elevation (end of previous timestep for the first iteration). Given the Surcharge Release value, the Gate Opening is determined by interpolation of the Gate Opening Table. The Gate Opening is not allowed to be reduced if the reservoir is in surcharge operations (Surcharge Release > 0). If the Gate Opening from the Curve is less that the current Gate Opening, the current Gate Opening is maintained, but a new Surcharge Release is determined given that gate opening by interpolation of the Gate Opening Curve. Otherwise, the Surcharge Release from the Induced Surcharge Curve is used.

The reservoir is then mass balanced using the Inflow (net) and Surcharge Release to determine the end of period Storage. If the Surcharge Release drops the reservoir below the Recession Target, a new Surcharge Release is determined that would drop the reservoir to exactly the Recession Target and the mass balance would resolve for the end of period Storage. Then an average Storage is determined using the beginning and ending period Storages. If the difference between the current storage and the new average storage is greater than the Storage Tolerance, the process is repeated using the new average storage to compute the new current pool elevation and a new Surcharge Release. Otherwise, the method moves on to the next forecast timestep.

17.1.21.4 Pass Inflows

This method is designed to pass the inflows to the reservoir whenever the pool elevation enters the surcharge pool. If the inflows cannot be passed (due to the physical limitations of the outlet works) the surcharge release is set equal to the max outflow. Then the gates remain open until the surcharge pool is evacuated.

Slots Specific to this Method

☛ Operating Level Table

Type: Periodic
Units: Time vs Length at Operating Levels
Description: table of describing the seasonal variation of elevation (storage) in a reservoir at each of the user-designated operating levels.
Information: number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next

timestep. See **HERE (Section 16.1.23)** for a method to modify which operating level table is used within a run.

I/O: Required Input
Links: Not Linkable

☛ Operating Level
Type: Series
Units: NONE
Description: The computed operating level
Information: This slot is computed using the pool elevation and the Operating Level Table
I/O: Output only
Links: Not Linkable

☛ Forecast period
Type: Scalar
Units: NO UNITS
Description: number of timesteps in the forecast period.
Information: number of timesteps over which inflows are forecasted and surcharge releases are calculated; includes current timestep.
I/O: Required Input
Links: Not Linkable

☛ Top of Flood Pool
Type: Scalar
Units: NO UNITS
Description: Operating level (as defined in Operating Level Table) corresponding to the top of flood pool.
I/O: Required Input
Links: Not Linkable

☛ Top of Conservation Pool
Type: Scalar
Units: NO UNITS
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the conservation pool.
I/O: Required Input
Links: Not Linkable

☛ Surcharge Release
Type: Series
Units: FLOW

Description: the computed surcharge release
Information: set by surcharge release method for all timesteps in forecast period
I/O: Output
Links: Not Linkable

This method is executed for every timestep in the forecast period when the surcharge release flag is set on the Outflow slot. The surcharge release is set equal to the total inflow to the reservoir (Inflow plus Hydrologic Inflow Forecast) if the pool elevation will exceed the top of the flood pool. If the inflow cannot be passed due to outlet constraints, the surcharge release is set equal to the maximum outflow. If this happens, the gates remain fully open and the surcharge release is set equal to the max outflow until the surcharge pool is evacuated. If the computed surcharge release will bring the pool elevation below the top of the flood pool, it will be reset to bring the pool elevation exactly to the top of the flood pool.

17.1.21.5 Specified Surcharge

This method is used to compute the surcharge release based on a user specified value. As always, the surcharge release is limited to the physical maximum outflow from the reservoir.

Slots Specific to this Method

☛ **Operating Level Table**
Type: Periodic
Units: Time vs Length at Operating Levels
Description: table of describing the seasonal variation of elevation (storage) in a reservoir at each of the user-designated operating levels.
Information: number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep. See HERE (Section 16.1.23) for a method to modify which operating level table is used within a run.
I/O: Required Input
Links: Not Linkable

☛ **Operating Level**
Type: Series
Units: None
Description: The computed operating level
Information: This slot is computed using the pool elevation and the Operating Level Table

I/O: Output only
Links: Not Linkable

☛ FORECAST PERIOD

Type: Scalar
Units: NO UNITS
Description: number of timesteps in the forecast period.
Information: number of timesteps over which inflows are forecasted and surcharge releases are calculated; includes current timestep.
I/O: Required Input
Links: Not Linkable

☛ TOP OF FLOOD POOL

Type: Scalar
Units: NO UNITS
Description: Operating level (as defined in Operating Level Table) corresponding to the top of flood pool.
I/O: Required Input
Links: Not Linkable

☛ TOP OF CONSERVATION POOL

Type: Scalar
Units: NO UNITS
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the conservation pool.
I/O: Required Input
Links: Not Linkable

☛ SURCHARGE RELEASE

Type: Series
Units: FLOW
Description: the computed surcharge release
Information: Set by surcharge release method for all timesteps in forecast period
I/O: Output
Links: Not Linkable

☛ SPECIFIED SURCHARGE RELEASE

Type: Scalar Slot
Units: FLOW
Description: The user input surcharge release

Information: This value is used to set the surcharge release slot whenever the pool elevation is in the surcharge pool but less than the Critical Elevation
I/O: Required Input
Links: Not Linkable

☛ Critical Elevation

Type: Scalar Slot
Units: Length
Description: the elevation at which the surcharge release changes from the Specified Surcharge Release to the maximum release
I/O: Required Input
Links: Not Linkable

This method is executed for every timestep in the forecast period when the surcharge release flag is set on the Outflow slot. The surcharge release is set equal to the Specified Surcharge Release if the pool elevation will exceed the top of the flood pool. If the pool elevation will exceed the Critical Elevation, the surcharge release is set equal to the maximum outflow. If this happens, the gates remain fully open and the surcharge release is set equal to the max outflow until the surcharge pool is evacuated. If the computed surcharge release will bring the pool elevation below the top of the flood pool, it will be reset to bring the pool elevation exactly to the top of the flood pool.

17.1.22 Rating Curves Modification

This category is dependent on having **Flat Top Surcharge** method selected in the **Surcharge Release** category. Methods are used to optionally modify the Rating Curves table.

17.1.22.1 None

This is the default method, no modification to the Rating Curves is made.

17.1.22.2 Specify Rating Curves using Elevation

When the **Specify Rating Curves using Elevation** method is selected the following slot is added:

☛ **RATING CURVES USING ELEVATION**

Type: Table Slot
Units: ELEVATION VS FLOW VS FLOW
Description: This table represents the Rating Curves Table with Elevation as the independent variable instead of Storage as shown on the Rating Curves table. Thus, The table relates Elevation, Induces Surcharge, and Free Flow Rating Curve.
Information: Specified only
I/O: Not linkable

When this method is selected, at the beginning of the run, the reservoir will generate the Rating Curves based on the Rating Curves using Elevation slot as follows:

1. Copy all of the data in the Rating Curves using Elevation table to the Rating Curves table. This will resize and/or rebuild the table as necessary.

2. For each row in the Rating Curves using Elevation table, the Elevation will be looked up on the Elevation Volume Table to determine the Storage.

3. The Storage will then be written to the Rating Curves table. The Induced Surcharge Curve and Free Flow Rating values were already copied from the Rating Curves using Elevation table.

4. The Rating Curves table will be marked as “has Source Slot”. This makes the slot read-only with cross hatching over the value. It also adds: “Note: these values are automatically generated based on Reservoir.Rating Curves using Elevation”.

Within the Flat Top Surcharge method, the original Rating Curves (storage based) tables will be used for all computations.

17.1.23 Elevation Max Duration Constraints

This category is dependent on the Flat Top Surcharge method being selected in the Surcharge Release category:

17.1.23.1 None

This is the default method in this category. There are no maximum duration constraints on Pool Elevation.

17.1.23.2 Constant Additional Surcharge Release

The Constant Additional Surcharge Release method allows the user to specify 1 or more elevation max duration constraints. The method is executed at the end of the Flat Top Surcharge method to compute the additional volume of water to meet the maximum elevation duration constraints. This volume is then released as a constant (additional) flow of water from the current timestep to the violation date.

The new method has two method specific slots:

☛ **ELEVATION MAXIMUM DURATION TABLE**

Type: Table Slot
Units: LENGTH VS NUMBER OF TIMESTEPS VS NUMBER OF TIMESTEPS
Description: The input data in this table drive this method.
Information: It has three columns: (1) Pool Elevation (length): the elevation to which the constraint applies. Rows must be in order of decreasing elevations. (2) Maximum Duration: the maximum number of timesteps this reservoir can be above this row's Pool Elevation. (3) Reset Duration: the minimum number of timesteps the reservoir elevation must remain at or below this row's Pool Elevation in order to reset the duration counter; i.e., how long must the Pool Elevation remain low to be considered not “above this row's Pool Elevation”.
I/O: Required Input
Links: NA

☛ **ELEVATION MAXIMUM DURATION RELEASE**

Type: Aggregate Series Slot
Units: FLOW
Description: The additional flows added to Surcharge Release explicitly for the purpose of meeting the Pool Elevation maximum duration constraints.
Information: The first column is the total for all constraints, the other columns correspond to single constraints. The column name are based on the row labels in the table described above.
I/O: Output only
Links: NA

This method is executed as part of the Flat Top Surcharge. After the Surcharge Release calculation has calculated the reservoir releases necessary for evacuating the surcharge pool over the forecast period, this method determines if this proposed release schedule would violate a Pool Elevation maximum duration constraint. If so, then the surcharge release is increased as necessary to avoid violating the constraint.

The additional flow is calculated as the volume of water which would need to be released to avoid violation of the constraint, and adding that volume to the release schedule in equal increments, i.e. constant additional release. This calculation is performed for each row of the Elevation Maximum Duration Table. Each iteration may add water to the proposed release.

Continuing with the example above, assume that at timestep t the reservoir computes the Surcharge Releases and that (given these releases) the reservoir's storage will have been between 800 and 900 m for 5 days at timestep t + 2. The difference in storage between the projected Pool Elevation at t + 2 and 1000 m is 300 m3. RiverWare will then add 100 m3 to the surcharge releases (and Outflow) for t, t + 1, and t + 2 to bring the elevation down to 1000 m at t + 2.

This method sometimes produces results that are counter-intuitive. Following are some items to consider when looking at results:

- **Multiple constraints lead to non constant release patterns:** This method determines the constant additional release required to meet each constraint. The actual additional flow that will be added to the surcharge release is the sum of the values for each constraint. Therefore, the final release schedule may not be a constant value if there are multiple constraints in effect.
- **Surcharge pattern leads to non constant release patterns:** The originally computed Surcharge Release schedule is not constant, so adding flows to the surcharge release will result in a non-constant release schedule.
- **Forecast values are overwritten at each timestep:** The Surcharge Release method, including this method, is executed at each timestep and will update the values in the Surcharge Release and other slots throughout the forecast period. Therefore, at the end of the run, the results may not be obvious what is happening. If you really want to see the schedules used for a particular timestep, pause the run after that timestep has executed, then look at slot values or the special results in the model Run Analysis tool, HERE (USACE_SWD.pdf, Section 5.5).
- **Constraining elevations may not be exactly met:** Surcharge Release operations do not include Evaporation or Precipitation in the computation. Similarly, reservoir diversions (withdrawals) are performed after surcharge operations. Once the other operations execute and the reservoir dispatches and solves, the resulting elevation may be slightly different than the values used in the Surcharge and elevation max duration computations. This will manifest in not exactly reaching the constraining elevations. Because

diversions (and evaporation) remove water from the reservoir, typically the elevations will be slightly below the constraining elevations.

- **Duration constraints still aren't met:** Maximum outflow constraints are applied after this additional release schedule is computed. Thus, even if the an additional amount is required, it may not be released and the target elevation may not be achieved. The proposed values are stored in the Surcharge Release slot, the actual maximums would be stored in the Outflow slot.

- **Elevations are unexpectedly at (or below) the constraint:** The algorithm looks at the pool elevation after forecasting and surcharge have executed. (This pre-constrained pool elevation is currently not available but looking at the forecasted inflows gives some indication of the values.) It then determines the additional flow to prevent violating the constraint at any time in the forecast period. If the elevation is already at the constraint, then the additional flow will likely keep the elevations at or even below the constraint. The method will add releases to the **current timesteps** (and future) timestep. There is no mechanism to delay releasing water to wait until the flood actually arrives. Thus, if it seems like the elevation should stay above the constraint, the algorithm will start releasing to make room for forecasted inflows.

  For example, when a large flood is within the period of perfect knowledge, then the algorithm may then need to release a large volume to meet a constraint at some point in the forecast period. This may cause the reservoir to go below the constraint to make room for the incoming flood (i.e. dig a hole for the incoming flood). But, the elevation should end up at the constraining elevation on the constraining timestep.

- **There are still violations in the forecast period:** Only one violation per constraint is considered, though in theory there could be another violation later in the forecast period.

17.1.24 Flood Control Release

Flood control releases on reservoirs cause dependent slots to be available for use by the predefined flood control rule function.

17.1.24.1 None

The default. No dependent slots.

17.1.24.2 Operating Level Balancing

Choose this method when you wish to use rules to perform Operating Level Balancing flood control on a computational subbasin (**HERE (Section 7.1.3.3)**) of which this reservoir is a member. Use of this method for USACE-SWD is described **HERE (USACE_SWD.pdf, Section 3.6)**.

SLOTS SPECIFIC TO THIS METHOD

☛ **FORECAST PERIOD**

Type: Scalar
Units: NO UNITS
Description: Number of timesteps in the forecast period.
Information: Number of timesteps over which storages are forecasted and surcharge releases are calculated; includes current timestep. Values must be identical for all reservoirs in the computational subbasin on which flood control is performed. May be propagated from the subbasin.
I/O: Required Input
Links: Not Linkable

☛ **BALANCE PERIOD**

Type: Scalar
Units: NO UNITS
Description: Number of timesteps in the balance period.
Information: Number of timesteps into the future at which the flood control inspects the storage of the reservoir to determine the goal for flood control releases. The goal is to release all the water in the flood pool at this future timestep (the current timestep is 1). Values must be identical for all reservoirs in the computational subbasin on which flood control is performed. May be propagated from the subbasin.
I/O: Required Input
Links: Not Linkable

☛ Target Balance Level

Type: Series
Units: No Units
Description: Balance level assigned to this subject reservoir by a controlling key control point, during execution of the control point's Operating Level Balancing method in the Key Control Point Balancing category.
Information: This slot is written as a result of the FloodControl predefined function HERE (RPLPredefinedFunctions.pdf, Section 40).
I/O: Output only - set by a rule
Links: Not Linkable

☛ Max Flood Control Release

Type: Series
Units: Flow
Description: Upper bound on flood control release, used by Operating Level Balancing flood control algorithm, and computed by Operating Level Balancing key control point balancing method on key control points. The flood control algorithm applies this upper bound on intermediate passes of the algorithm, but not on the last pass.
Information: This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Rule output.
Links: Not linkable.

☛ Flood Control Release

Type: Series
Units: Flow
Description: The release made from the reservoir as a result of the application of a flood control rule.
Information: The value for this slot is computed by the flood control predefined rules function. The slot's value is meant to be assigned by the rule.
I/O: Rule output.
Links: Not Linkable

☛ Flood Control Minimum Release

Type: Series Slot
Units: Flow
Description: This slot holds the user specified (via a rule) minimum flood control release. This slot is used by the Operating Level Balancing algorithm as a minimum release, but is not included in the Flood Control Release slot's value. Click HERE (USACE_SWD.pdf, Section 3.4) for more information on the intended use of this slot.

Information: This value should be set (along with Outflow) prior to calling the flood control method so that water is allowed to route downstream.
I/O: Typically set by a rule
Links: Not Linkable

☛ Proposed Flood Control Release

Type: Series Slot
Units: FLOW
Description: The temporary running value of the flood control release calculated by the Flood Control Release method on the Computation Subbasin.
Information: This slot is used to hold "proposed" flood control release during the flood control calculations. This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output
Links: Not linkable

☛ Surcharge Release

Type: Series
Units: Flow
Description: The release made from the reservoir as a result of the application of surcharge release rule.
Information: The values in this slot are used by the Operating Level Balancing flood control method invoked by the predefined rules function.
I/O: Rule output.
Links: Not Linkable

☛ Top of Flood Pool

Type: Scalar
Units: No Units
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the flood pool.
Information: Values must be identical for all reservoirs in the computational subbasin on which flood control is performed. May be propagated from the subbasin.
I/O: Required Input
Links: Not Linkable

☛ Top of Conservation Pool

Type: Scalar
Units: No Units
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the conservation pool.

Information: Must be lower than the Top of Flood Pool. Values must be identical for all reservoirs in the computational subbasin on which flood control is performed. May be propagated from the subbasin.
I/O: Required Input
Links: Not Linkable

☛ Operating Level Table

Type: Periodic
Units: Time vs Length at Operating Levels
Description: Describes the seasonal variation of elevation (storage) in a reservoir at each of the user-designated operating levels.
Information: Number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep. See HERE (Section 16.1.23) for a method to modify which operating level table is used within a run.
I/O: Required Input
Links: Not Linkable

☛ Allowable Rising Release Change

Type: Scalar
Units: Flow Per Time
Description: The maximum acceleration of the flood control releases, used by the rulebased Operating Level Balancing flood control. Value must be greater than 0.
Information: The Operating Level Balancing flood control algorithm attempts not to exceed this increase in outflow from one timestep to the next when computing flood control releases. Surcharge releases may cause this change to be exceeded. Value is converted to the internal units appropriate to the timestep size of the run, thus, is independent of the run's timestep size.
I/O: Required Input
Links: Not Linkable

☛ Allowable Falling Release Change

Type: Scalar
Units: Flow Per Time

Description: The maximum deceleration of the flood control releases, used by the rulebased Operating Level Balancing flood control. Value must be greater than 0.

Information: The Operating Level Balancing flood control algorithm attempts not to exceed this reduction in outflow from one timestep to the next when computing flood control releases. Channel space limitations may cause this change to be exceeded. Value is converted to the internal units appropriate to the timestep size of the run, thus, is independent of the run's timestep size.

I/O: Required Input

Links: Not Linkable

☛ Maximum Release Variation

Type: Scalar Slot

Units: Flow Per Time

Description: A deceleration of the flood control releases assumed to occur during calculation of channel space in the rulebased Operating Level Balancing flood control. Value must be greater than 0.

Information: The Operating Level Balancing flood control algorithm uses this value as a tuning parameter. Large values mean that flood releases may be larger "today" and thus, zero "tomorrow" due to routing lags, whereas small values may mean that less-than-ideal releases can be made "today"; in either case, poor choices can cause oscillations in the reservoirs' releases. Value is converted to the internal units appropriate to the timestep size of the run, thus, is independent of the run's timestep size.

I/O: Required Input

Links: Not Linkable

☛ Through Release

Type: Series

Units: Flow

Description: Water released from upstream reservoirs that flows through this reservoir. This reservoir is a downstream tandem.

Information: This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool **HERE (USACE_SWD.pdf, Section 5.5)**.

I/O: Output only

Links: Not Linkable

☛ Tandem Storage

Type: Series

Units: Flow

Description: All or part of a proposed release may be stored in downstream "tandem" reservoirs if the downstream reservoir is not Surcharging (Surcharge Release equals zero or is not valid) at the current controller timestep. This slot tracks the water stored in this reservoir.

Information: This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool **HERE (USACE_SWD.pdf, Section 5.5)**.

I/O: Output only

Links: Not Linkable

☛ Lost Tandem Storage

Type: Agg Series

Units: FLOW

Description: This slot tracks the lost storage (as a flow) that occurs because tandem releases are not routed from the upstream to the downstream reservoir. It is really an error term. The first column is total of the other columns. Colum2 is for the current timestep releases. Column 2 is for timestep 1 through the balance period releases. Column 3 is for releases from the balance period through forecast period. For example the code decides to release 100cfs that can be stored in the downstream tandem. If you don't route it, 100 gets stored on the current timestep. But if you had routed it, some fraction would not have made it there on day 1 (1st column). If you instead propose a hydrograph of flows that don't get routed, then you can compute the col 2 and col 3 values.

Information: This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool **HERE (USACE_SWD.pdf, Section 5.5)**.

I/O: Output Only

? Downstream Control Point Share

Type: Agg Series

Units: DECIMAL

Description: The slot has a series slot for each downstream key control point. The value in the slot shows the share as presented on the downstream Control Point's Share slot. Essentially, this slot contains a copy of the downstream Control Point's Share information.

Information: This slot is invisible but can be viewed in the reservoir's Special Results details on the Model Run Analysis tool **HERE (USACE_SWD.pdf, Section 5.5)**.

I/O: Output only

Links: Not Linkable

17.1.24.3 Phase Balancing

Choose this method when you wish to use rules to perform Phase Balancing flood control on a computational subbasin of which this reservoir is a member. See Phase Balancing discussion in the Computational Object for a complete discussion of the Phase Balancing algorithm.

Slots Specific to this Method

☛ Forecast period

Type: Scalar
Units: No Units
Description: Number of timesteps in the forecast period.
Information: Minimum value of 1. This value should be the same as the Forecast Period on other objects in the subbasin; the value can optionally be propagated from the subbasin.
I/O: Required Input
Links: Not Linkable

☛ Surcharge Release

Type: Series
Units: Flow
Description: release required by operating policy when the reservoir elevation is in the surcharge pool.
Information: set by surcharge release method for all timesteps in forecast period when "Surcharge Release" flag is set on this slot by end user or by a rule
I/O: Output
Links: Not Linkable

☛ Top of Flood Pool

Type: Scalar
Units: No Units
Description: operating level (as defined in Operating Level Table) corresponding to the top of flood pool
Information: KC COE has used 4 in the past, but can be any number
I/O: Required Input
Links: Not Linkable

☛ Top of Conservation Pool

Type: Scalar
Units: nONE
Description: The operating level pool number of the top of conservation pool level.
Information: This value is also termed the Target Operating Pool Level, since this is the preferred, or "Target" level for all reservoirs. This level is also the bottom of the flood pool level.
I/O: Input
Links: Not linkable

☛ Operating Levels Table

Type: Periodic
Units: TIME vs Length at Operating levels
Description: table describing the seasonal variation of elevation (storage) in a reservoir at each of the user-designated operating levels.
Information: number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep.
I/O: Required Input
Links: Not Linkable

☛ Operating Level

Type: Series
Units: No units
Description: Operating level of the reservoir at each timestep, based on the ending Storage.
Information: Determined from interpolation of the Operating Levels Periodic plus the delta from the Operating Levels Aberration Agg Series; an exact value is determined, not limited to an integer or values entered in the Operating Level Table.
I/O: Output
Links: Not Linkable

☛ Phase Tolerance

Type: Scalar
Units: No Units
Description: The tolerance value that must be exceed for the reservoir to change phase.
Information: The reservoir phase is detirmined from the operating level. The operating level must exceed a phase boundary by the “Phase Tolerance” before the lake will change phase. This slot can be used to smooth oscillations along phase boundaries.
I/O: Input (0.0 default)
Links: Not Linkable

☛ Lake Character

Type: Series
Units: No units
Description: weighting factor for allocating available downstream channel space.

Information: used by control point or flood control algorithm to balance reservoirs in calculating flood releases
I/O: Output
Links: Not Linkable

☛ Lake Character Coefficient

Type: Scalar
Units: No UNITS
Description: coefficient multiplied by the occupied volume of the flood pool to determine the Lake Character.
Information: used for every timestep, unless Variable Lake Character Coefficient has been input (not a NaN) for some timestep(s).
I/O: Input
Links: Not Linkable

☛ Variable Lake Character Coefficient

Type: Series
Units: NO UNITS
Description: time-varying Lake Character Coefficient
Information: replaces Lake Character Coefficient Scalar when input for some timestep(s).
I/O: Optional Input
Links: Not Linkable

☛ Flood Control Release

Type: Series
Units: FLOW
Description: value of the flood control release calculated by the Flood Control Release method and set by the rule that calls that method.
Information: During the flood control calculations at each timestep, this value is calculated for the entire forecast period; at the next timestep, the forecast values are overwritten by the new values. At the end of the run, this slot contains the flood control release for all the timesteps. During dispatching, this value is added to the surcharge release and possibly other values to get the total outflow for the timestep.
I/O: Output
Links: Not linkable

☛ Proposed Flood Control Release

Type: Series
Units: FLOW
Description: The temporary running value of the flood control release calculated by the Flood Control Release method on the Computation Subbasin.

Information: This slot is used to hold “proposed” flood control release during the flood control calculations. This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output
Links: Not linkable

☛ Recalc Release

Type: Series
Units: FLOW
Description: The temporary running value of the reservoir’s current constrained release.
Information: This slot is used to hold the last constrrained release of the reservoir. This value is kept in case the computational subbasin has the opportunity to recalculate the reservoir’s release (i.e., one of this reservoir’s siblings cannot use all of its allocation allowing this flood control release of this reservoir to be increased). This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output
Links: Not linkable

☛ Known Release

Type: Series
Units: FLOW
Description: The temporary running value of the reservoir’s “known” or solved for releases.
Information: This slot is used for bookkeeping, holding the values of “known” or solved for releases in the forecast period during the flood control release algorithm. This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output
Links: Not linkable

☛ Phase

Type: Series
Units: No Units
Description: A temporary record of the lake’s phase.
Information: This slot is used for bookkeeping, holding the values of the lake phase. This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output)
Links: Not Linkable

☛ Minimum Release

Type: Periodic
Units: FLOW
Description: The minimum release the reservoir is required to make. A seasonal value.
Information: number of rows defined by the number of date points (user input) and one column. The column represents the time-varying minimum release for the reservoir.elevations for a particular Operating Level. An flow value is input for each date point. User can select whether to interpolate between values in time, or to have constant values until the next timestep.
I/O: Required Input
Links: Not Linkable

☛ Tandem Operating Levels Table

Type: Periodic
Units: TIME vs Length at Operating levels
Description: Table describing the seasonal variation of elevation (storage) in a tandem reservoir at each of the user-designated operating levels.
Information: number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep.
I/O: Required Input
Links: Not Linkable

☛ Tandem Operating Level Aberrations

Type: Agg Series Slot
Units: No units
Description: Aberration of the operating levels of the tandem reservoir from the periodic specification at each timestep
Information: The slot contains an aberration series for each operating level. The abberation is used to adjust the tandem operating levels when constructing the tandem balancing curve.
I/O: Input
Links: Not Linkable

☛ Objective Release Pattern Table

Type: Table
Units: No Units

Description: Table describing the objective releases percentages used to evacuate the flood control storage (including forecasted inflows) over the period of the table from the first unconstrained release.
Information: The n^{th} number in the table represents the percentage of the flood control storage (include forecasted inflows) that should be released for the n^{th} day from the first unconstrained release from this reservoir.
I/O: Required Input
Links: Not Linkable

☛ Maximum Objective Release
Type: Series
Units: FLOW
Description: The temporary running values of the reservoir's current objective releases. This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output
Links: Not linkable

☛ Objective Pattern Threshold
Type: Table
Units: NONE
Description: Percentage of allowable volume change in objective release pattern
Information: This is a Scalar representing the percentage of volume changed. If the change volume of flood control storage in a reservoir (including forecasted inflows) is within the threshold of the objective release pattern, then the pattern can be maintained and does not need to be recomputed. If the change exceeds the threshold then the pattern must be recomputed. The table has two columns. The first column lists outflows. The second column list the corresponding permissible decrease in outflow.
I/O: Required Input
Links: Not Linkable

☛ Permissible Outflow Increase Constraints Table
Type: Table
Units: FLOW, FLOW
Description: Table describing the allowable outflow increase given a current outflow.
Information: The table has two columns. The first column lists outflows. The second column list the corresponding permissible increase in outflow.
I/O: Required Input
Links: Not Linkable

☛ Permissible Outflow Decrease Constraints Table

Type: Table
Units: FLOW, FLOW
Description: Table describing the allowable outflow decrease given a current outflow.
Information: The table has two columns. The first column lists outflows. The second column list the corresponding permissible decrease in outflow.
I/O: Required Input
Links: Not Linkable

17.1.25 Bank Storage

The Bank Storage methods are used to calculate the volume of water stored in the Reservoir banks. These methods also calculate the change in the volume of water stored in the Reservoir banks from one timestep to the next.

17.1.25.1 None

None should be chosen if the user does not want to calculate the amount of Bank Storage in the Reservoir. This is the default method for the Bank Storage category. Bank Storage and the Change in Bank Storage are set to zero but are not displayed. There are no slots specifically associated with this method. No calculators are performed in this method.

17.1.25.2 Input Bank Storage

The Input Bank Storage method allows users to directly input values into the Bank Storage slot or to set these values using a rule. Change in Bank Storage is calculated internally in RiverWare for use in the mass balance equations.

SLOTS SPECIFIC TO THIS METHOD

☛ **BANK STORAGE**

Type: Series Slot
Units: VOLUME
Description: volume of water stored in the reservoir banks
Information:
I/O: Input Only
Links: Usually not linked, but could be linked to Data Object.

☛ **CHANGE IN BANK STORAGE**

Type: Series Slot
Units: VOLUME
Description: change in volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

17.1.25.3 CRSS Bank Storage

The CRSS Bank Storage method replicates the U.S. Bureau of Reclamation’s CRSS bank storage calculation. The Bank Storage and the Change in Bank Storage are calculated using the Reservoir Storage and the Bank Storage Coefficients.

SLOTS SPECIFIC TO THIS METHOD

☛ Bank Storage

Type: Series Slot
Units: Volume
Description: volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

☛ Bank Storage Coefficient

Type: Table Slot
Units: No Units vs. No Units
Description: gain or loss of storage vs. change in bank storage
Information: The first coefficient (column zero) is for increasing storage and the second coefficient is for decreasing storage.
I/O: Required input
Links: Not linkable

☛ Change in Bank Storage

Type: Series Slot
Units: Volume
Description: change in volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

There are two ways Bank Storage can be calculated depending on the current Storage of the Reservoir. If the Reservoir's current Storage is greater than the Reservoir's Storage at the previous timestep, the Storage is increasing. Bank Storage is calculated using the following equation:

$$\text{Bank Storage} = \text{Bank Storage}(-1) + (\text{first Bank Storage Coefficient} \times (\text{Storage} - \text{Storage}(-1)))$$

If the Reservoir's current Storage is less than the Reservoir's Storage at the previous timestep, the Storage is decreasing. Bank Storage is calculated using the following equation:

$$\text{Bank Storage} = \text{Bank Storage}(-1) + (\text{second Bank Storage Coefficient} \times (\text{Storage} - \text{Storage}(-1)))$$

The Change in Bank Storage is calculated using the following equation regardless of which method was used to compute Bank Storage.

$$\text{Change in Bank Storage} = \text{Bank Storage} - \text{Bank Storage}(-1)$$

where in the above equations:

Bank Storage = the volume of water stored in the banks of the Reservoir at the current timestep

Bank Storage(-1) = the volume of the water stored in the banks of the Reservoir at the previous timestep.

Storage = the volume of water in the Reservoir at the current timestep

Storage(-1) = the volume of water in the Reservoir at the previous timestep

17.1.25.4 Average Stage Change

The Average Stage Change method calculates the Bank Storage and Change in Bank Storage based on the flow from storage. The flow from storage is a function of the average stage change over a user defined number of timesteps.

Slots Specific to this Method

☛ **Ave Stage Change Coeffs**

Type: Table Slot
Units: Area per Time and Flow
Description: coefficient describing flow for a given change in pool elevation and a constant representing flow from bank storage
Information:
I/O: Required input
Links: Not linkable

☛ **Bank Storage**

Type: Series Slot
Units: Volume
Description: volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

☛ **Change in Bank Storage**

Type: Series Slot
Units: Volume
Description: change in volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

☛ TIMESTEPS TO AVERAGE

Type: Table Slot
Units: NO UNITS
Description: number of timesteps used to calculate average pool elevation.
Information:
I/O: Required input
Links: Not linkable

The average stage change is calculated using the following equation:

$$\text{average Pool Elevation} = \frac{\text{Pool Elevation} - \text{Pool Elevation (-Timesteps to Average)}}{\text{Timesteps to Average}}$$

The change in flow to bank storage is calculated using the following equation:

$$\text{Flow to banks} = \text{Average Stage Change Bank Storage Coefficient} \times \text{Average Pool Elevation} + \text{Average Stage Change Bank Storage Constant}$$

The flow is converted to a volume by multiplying the value by the current timestep. The Change in Bank Storage is calculated using the following equation:

$$\text{Change in Bank Storage} = \text{Bank Storage} - \text{Bank Storage(-1)}$$

17.1.26 Diversion from Reservoir

The Diversion from Reservoir user methods are applicable when a reservoir is linked to a diverting object (e.g. AggDiversionSite, AggDistributionCanal, or Diversion Object). These methods simply create the slots which must be linked (by the user) to slots on the diverting object.

17.1.26.1 None

This is the default for the Diversion from Reservoir category. It is used when the reservoir is not linked to a diverting object. If the reservoir is linked to a diverting object and this method is selected, the object will not solve correctly. There are no slots specifically associated with this method.

17.1.26.2 Available Flow Based Diversion

This method must be selected when a reservoir is linked to either an AggDiversionSite, AggDistributionCanal, or a Diversion Object that is using the Available For Diversion Linked method. Selecting this method allows the Available for Diversion slot to be available for linking. The AggDiversionSite, AggDistributionCanal, and Diversion objects contain more information about diverting water from a reservoir.

SLOTS SPECIFIC TO THIS METHOD

☛ **AVAILABLE FOR DIVERSION**

Type: Series
Units: FLOW
Description: represents the amount of water that may be diverted from the reservoir
Information:
I/O: Optional; can be input by the user or determined by **RiverWare**™.
Links: Should be linked to the Available for Diversion slot on AggDiversionSite or Diversion object, or the Incoming Available Water slot on a Water User.

Available for Diversion can either be input by the user or calculated by the reservoir. If it is not input it is set as the previous Storage divided by the timestep length. The value is limited to not be negative.

No other calculations are performed if this method is selected.

17.1.26.3 Head Based Diversion

This method may be selected when a reservoir is linked to a Diversion Object. Selecting this method allows the Previous Pool Elevation slot to be available for linking. The Diversion Object contains more information about diverting water from a Reservoir.

Slots Specific to this Method

☛ Previous Pool Elevation

Type: Series
Units: Length
Description: Pool Elevation value for the previous timestep
Information:
I/O: Output only
Links: Should be linked to the Diversion Intake Elevation slot on the Diversion Object.

17.1.27 Diversion Power

The methods in this category calculate power generated on the diversion from the reservoir. Selecting a method other than None in this category will make the Diversion Tailwater and Diversion Power Bypass categories available. If a method other than None is selected for Diversion Power, then a method other than None must be selected for Diversion Tailwater.

17.1.27.1 None

This is the default method for the Diversion Power category. No calculations are performed in this method, and there are no slots specifically associated with this method.

17.1.27.2 Diversion Power Efficiency Curve

The Diversion Power Efficiency Curve method is similar to the Plant Efficiency Curve method in the Power category with the exception that the method does not allow **Diversion Energy** to input or set by rules (nor can **Diversion Energy** be set with the Best Efficiency or Max Capacity flags). **Diversion Energy** and **Diversion Power** are only calculated as outputs. The method calculates **Diversion Power** by a 3-D interpolation of the **Diversion Power Table** using the current, average **Diversion Operating Head** and **Diversion Turbine Flow**. The **Diversion Power Coefficient** is calculated as **Diversion Power** divided by **Diversion Turbine Flow**. Alternatively, the user can input **Diversion Power Coefficient**, and then **Diversion Power** is calculated directly as the **Diversion Power Coefficient** multiplied by the **Diversion Turbine Flow**.

Slots Specific to this Method

☛ **Diversion Power Table**

Type: Table Slot
Units: Length vs Flow vs Power
Description: 3-D table representing the power characteristics of the diversion power plant, used to calculate power using interpolation
Information: Data must be entered into the table in increasing, blocks of the same Diversion Operating Head. For every block of the same Diversion Operating Head in column 1, Diversion Turbine Flow should be listed in increasing order in column 2 and the corresponding Diversion Power in column 3. The first row for each Diversion Operating Head must be for zero Diversion Turbine Flow and zero Diversion Power.
I/O: Required input
Links: Not linkable

Div Head	Turbine Flow	Div Power
30	0	0
30	100	100
30	200	175
40	0	0
40	100	125
40	220	195
50	0	0
50	110	147
50	250	205

☛ Diversion Max Turbine Table

Type: Table Slot
Units: Length vs Flow
Description: The maximum Diversion Turbine Flow as a function of Diversion Operating Head
Information: RiverWare automatically populates this table at the start of the run using the Diversion Power Table. The first column contains the Diversion Operating Head values from the Diversion Power Table, one row for each unique Diversion Operating Head in increasing order. The second column contains the maximum Diversion Turbine Flow value for each Diversion Operating Head.
I/O: Output only
Links: Not Linkable

☛ Diversion Power Cap Fraction

Type: Series Slot
Units: Fraction
Description: This is the percentage of full capacity of the turbine units in the diversion power plant. For example, if only half of the turbine are operational (and they are all the same), this value would be 0.5.
Information: This must be a number between 0 and 1 (inclusive). If not input or set by rules, this slot is automatically set to 1.
I/O: Optional input, if not, value is set to 1
Links: Not linkable

☛ Diversion Operating Head

Type: Series Slot

Units: LENGTH
Description: The difference between the average Pool Elevation and the Diversion Tailwater Elevation
Information:
I/O: Output only
Links: Not usually linked

DIVERSION TURBINE FLOW

Type: Series Slot
Units: FLOW
Description: The diversion flow that passes through the turbines to generate power
Information: If the slot is not input or set by rules, then it is calculated as the difference between Diversion and Diversion Power Bypass if Diversion Power Bypass is input or set by rules. If neither Diversion Turbine Flow nor Diversion Power Bypass is input or set by rules, then Diversion Turbine Flow is calculated as the lesser of Diversion and the calculated maximum diversion turbine flow based on the Diversion Max Turbine Table and the current Diversion Operating Head. It is not permissible to have both Diversion Turbine Flow and Diversion Power Bypass as input or set by rules.
I/O: Optional input or output
Links: Not linkable

DIVERSION POWER

Type: Series Slot
Units: POWER
Description: The power generated from flow through the reservoir diversion
Information: If Diversion Power Coefficient is not input or set by rules, Diversion Power is calculated using a 3-D interpolation on the Diversion Power Table given the current, average Diversion Operating Head and the current Diversion Turbine Flow, scaled by the Diversion Power Cap Fraction. If Diversion Power Coefficient is input or set by rules, Diversion Power is calculated as the Diversion Power Coefficient multiplied by the Diversion Turbine Flow.
I/O: Output only
Links: Linkable

DIVERSION ENERGY

Type: Series Slot
Units: ENERGY
Description: The energy generated from flow through the reservoir diversion
Information: Calculated as the Diversion Power multiplied by the timestep length
I/O: Output only
Links: Linkable

DIVERSION POWER COEFFICIENT

Type: Series Slot
Units: POWER PER FLOW
Description: The power generation per unit of flow through the turbines on the reservoir diversion
Information: If this slot is input or set by rules, it is used directly to calculate Diversion Power. If it is not input or set by rules, then it is calculated as Diversion Power divided by Diversion Turbine Flow. If either Diversion Power, Diversion Turbine Flow or Diversion Power Cap Fraction is zero, then this slot will be zero.
I/O: Optional input or output
Links: Not usually linked

At the start of the run, the **Diversion Max Turbine Table** slot is populated using the **Diversion Power Table**. The first column is populated with each unique **Diversion Operating Head** value from the **Diversion Power Table**, in ascending order. The second column is populated with the corresponding maximum diversion turbine flow value.

When the method executes, **Diversion** will already be known. The method calls the selected Diversion Tailwater method to calculate the **Diversion Tailwater Elevation**. If the default method, **None**, is selected for the **Diversion Tailwater** category, the run will abort and an error message will be issued.

The method then calculates the **Diversion Operating Head**:

$$\text{Diversion Operating Head}[t] = \frac{\text{Pool Elevation}[t-1] + \text{Pool Elevation}[t]}{2} - \text{Diversion Tailwater Elevation}[t]$$

The method calculates the *maxDiversionTurbineFlow* by interpolating the **Diversion Max Turbine Table** slot using the **Diversion Operating Head**. The value is scaled by the **Diversion Power Cap Fraction**.

If **Diversion Turbine Flow** is specified (input or set by rules) it is checked against the *maxDiversionTurbineFlow*, and if the specified value exceeds the max, the run will abort with an error message. Otherwise a temporary turbine flow is calculated.

$$\text{tempDiversionTurbine} = Min(\text{Diversion} - \text{Diversion Power Bypass}, maxDiversionTurbineFlow)$$

If **Diversion Power Bypass** is not input or set by rules, or if the **Diversion Power Bypass** method is **None**, the **Diversion Power Bypass** will have defaulted to zero at this point.

If the combined temporary turbine flow plus the current **Diversion Power Bypass** is less than the (total) **Diversion**, then the method then calls the selected Diversion Power Bypass method to increase the **Diversion Power Bypass** to make up the difference. If it is not possible for the turbine flow plus the bypass to equal the total **Diversion**, either due to the values being specified (input or rules) or due to max capacity limits, then the run will abort with an error message.

The method then sets the Diversion Turbine Flow slot:

$$\text{Diversion Turbine Flow} = \text{tempDiversionTurbine}$$

If **Diversion Power Coefficient** is specified (input or rules), it is used to calculate **Diversion Power** directly:

$$\text{Diversion Power} = \text{Diversion Turbine Flow} \times \text{Diversion Power Coefficient}$$

Otherwise, **Diversion Power** is calculated by a 3-D interpolation on the **Diversion Power Table** using **Diversion Operating Head** and **Diversion Turbine Flow**, and **Diversion Power Coefficient** is calculated as:

$$\text{Diversion Power Coefficient} = \frac{\text{Diversion Power}}{\text{Diversion Turbine Flow}}$$

Diversion Energy is then calculated as **Diversion Power** multiplied by the timestep length.

Notes on Diversion Power Cap Fraction

If the **Diversion Power Cap Fraction** is input by the user, it is necessary for the **Diversion Power Table** to be scaled back to account for the operating points when the turbines are operating at less than 100%. To do this, **Diversion Turbine Flow** is divided by the **Diversion Power Cap Fraction**. This point is then found in the **Diversion Power Curve** for the current **Diversion Operating Head**, and the power is found using 3-D interpolation. Finally the power is multiplied by the **Diversion Power Cap Fraction** to get the actual **Diversion Power** produced for the current timestep.

17.1.28 Diversion Tailwater

The methods in this category calculate the elevation of the tailwater on the diversion from a reservoir. This category is dependent on the selection of a method other than the default method, None, in the Diversion Power category. If a method other than the default is selected for Diversion Power, then a method other than the default, None, must be selected for Diversion Tailwater.

17.1.28.1 None

This is the default method for the Diversion Tailwater category. No calculations are performed in this method, and there are no slots specifically associated with this method.

17.1.28.2 Diversion Base Value Plus Lookup

The Diversion Base Value Plus Lookup method computes the **Diversion Tailwater Elevation** by added the average **Diversion Tailwater Base Value** (over the timestep) to a function of **Diversion** defined in the **Diversion Tailwater Table** slot. This method is similar to the Base Value Plus Lookup Table method in the Tailwater category but uses the **Diversion** and **Diversion Tailwater Base Value** slots instead of **Outflow** and **Tailwater Base Value**. The **Diversion Tailwater Base Value** may be input by the user or linked to another slot, such as the **Pool Elevation** of another Reservoir. If the **Tailwater Base Value** is neither input nor linked, it is automatically set to zero.

Slots Specific to this Method

☛ **Diversion Tailwater Table**

Type: Table Slot
Units: Flow vs Length
Description: This slot defines the relationship between Diversion and the Diversion Tailwater Elevation; Diversion vs either the diversion tailwater elevation or the tailwater elevation increment
Information: If the Diversion Tailwater Base Value is non-zero, the Diversion Tailwater Table gives values of incremental increase in Tailwater Elevation over th base value. Otherwise, the table gives the Diversion Tailwater Elevation values. The first row of the table should be for a Diversion flow of zero.
I/O: Required input
Links: Not linkable

☛ **Diversion Tailwater Base Value**

Type: Series Slot
Units: Length
Description: the base elevation of the diversion tailwater, such as a downstream stage

Information: If the slot is not input or linked, it defaults to 0.
I/O: Optional, can be input or linked
Links: Linkable

Diversion Tailwater Elevation

Type: Series Slot
Units: Length
Description: the water surface elevation of the tailwater from the reservoir diversion
Information: This slot is used to compute Diversion Operating Head in Diversion Power calculations
I/O: Output only
Links: Not linkable

When this method is executed, the **Diversion** value will already be known. If the **Diversion Tailwater Base Value** is neither linked, input nor set by rules, then it will default to zero.

The following steps are performed to calculate **Diversion Tailwater Elevation**.

1. *TWTemp* is obtained from a table interpolation on the **Diversion Tailwater Table** using **Diversion**.

2. If both **Diversion Tailwater Base Value[t]** and **Diversion Tailwater Base Value[t-1]** are known, then the **Diversion Tailwater Elevation** is calculated as:

$$\text{Diversion Tailwater Elevation}[t] = \frac{\text{Diversion Tailwater Base Value}[t-1] + \text{Diversion Tailwater Base Value}[t]}{2} + TWTemp$$

3. If **Diversion Tailwater Base Value[t]** is known, but **Diversion Tailwater Base Value[t-1]** is not known, then the **Diversion Tailwater Elevation** is calculated as:

$$\text{Diversion Tailwater Elevation}[t] = \text{Diversion Tailwater Base Value}[t] + TWTemp$$

4. If **Diversion Tailwater Base Value[t-1]** is known, but **Diversion Tailwater Base Value[t]** is not known, then the **Diversion Tailwater Elevation** is calculated as:

$$\text{Diversion Tailwater Elevation}[t] = \text{Diversion Tailwater Base Value}[t-1] + TWTemp$$

5. If neither **Diversion Tailwater Base Value[t]** nor **Diversion Tailwater Base Value[t-1]** are known but **Diversion Tailwater Elevation[t-1]** is known, the current timestep's **Diversion Tailwater Elevation** is set equal to **Diversion Tailwater Elevation[t-1]**.

6. If neither **Diversion Tailwater Base Value[t]**, **Diversion Tailwater Base Value[t-1]**, nor **Diversion Tailwater Elevation[t-1]** are known, or if **Diversion** is not known, the method will exit and wait for more information.

17.1.29 Diversion Power Bypass

The methods in this category calculate the portion of the diversion from a reservoir that does not pass through the turbines but rather through a bypass structure. This category is dependent on the selection of a method other than the default method, None, in the Diversion Power category.

17.1.29.1 None

This is the default method for the Diversion Power Bypass category. No calculations are performed in this method, and there are no slots specifically associated with this method. If this method is selected, it is assumed that all **Diversion** flow passes through the turbines.

17.1.29.2 Bypass Capacity Table

This method sets **Diversion Power Bypass** to the difference between **Diversion** and **Diversion Turbine Flow** if it is not input or set by rules, and it checks that the **Diversion Power Bypass** does not exceed the maximum based on the **Diversion Power Bypass Table**. This functions similarly to the Regulated method in the Spill category.

Slots Specific to this Method

☛ Diversion Power Bypass Table

Type: Table Slot
Units: Length vs Flow
Description: Pool Elevation vs. the corresponding maximum diversion power bypass values
Information:
I/O: Required input
Links: Not linkable

☛ Diversion Power Bypass

Type: Series
Units: Flow
Description: Diversion flow that does not pass through power turbines
Information: If not input or set by rules, Diversion Power Bypass will be set equal to the difference between Diversion and Diversion Turbine Flow.
I/O: Optional input or output
Links: Linkable

At the beginning of the run, if **Diversion Power Bypass** is not specified (input or rules), it is initially set to a default value of zero.

On each timestep, the method first checks if both **Diversion Power Bypass** and **Diversion Turbine Flow** are input or set by rules. If so, the run will abort with an error message. It necessary to leave at least one of these slots as a free variable.

The method then calculates max diversion power bypass by performing a table interpolation on the **Diversion Power Bypass Table** using the average **Pool Elevation** from the end of the current timestep and end of the previous timestep.

If **Diversion Power Bypass** is input or set by a rule, then the value is checked against the max diversion power bypass, and if it exceeds the max, the run will abort with an error message.

If **Diversion Power Bypass** is not input or set by a rule, then it is calculated as:

$$\text{Diversion Power Bypass} = \text{Diversion} - \text{Diversion Turbine Flow}$$

The calculated **Diversion Power Bypass** value is checked against the max diversion power bypass, and if it exceeds the max, the run will abort with an error message.

17.1.30 Seepage

The Seepage methods are used to calculate the amount of water lost through the face of the dam. The volume of seepage computed during the execution of these methods affects the mass balance of the Reservoir.

17.1.30.1 None

None is the default for the Seepage category. It is used when the user does not want to calculate the flow of water through the face of the dam.

17.1.30.2 Input Seepage

The Input Seepage method is used when it is desired to have the seepage slot as input or set by a rule.

SLOTS ADDED BY THIS METHOD:

☛ **SEEPAGE**

Type: SeriesSlot
Units: FLOW
Description: flow of water through the dam face
Information: Seepage is not included in the Outflow of the reservoir and will need to be linked separately if the water does in fact go downstream.
I/O: Input only
Links: Linkable

17.1.30.3 Linear Seepage

The Linear Seepage method calculates the seepage from the face of the dam. This calculation is based on the Pool Elevation of the Reservoir and specified coefficients.

SLOTS ADDED BY THIS METHOD:

☛ **SEEPAGE**

Type: SeriesSlot
Units: FLOW
Description: flow of water through the dam face
Information: Seepage is not included in the Outflow of the reservoir and will need to be linked separately if the water does in fact go downstream.
I/O: Output only
Links: Linkable

☛ **SEEPAGE COEFFICIENTS**

Type: TableSlot
Units: LENGTH, AREA PER TIME, FLOW
Description: coefficients in the linear equation for seepage
Information: The first coefficient (column zero) is the base elevation of the dam. The second coefficient is the slope of the linear equation for seepage. The third coefficient is the intercept of the linear equation for seepage.
I/O: Required Input
Links: NA

The calculation for Seepage in this method is fairly straightforward. A linear model is used. The coefficient are user inputs. The following equation is used to compute Seepage:

$$\text{Seepage} = (\text{Pool Elevation} - \text{first Seepage Coefficient}) \times \text{second Seepage Coefficient} + \text{third Seepage Coefficient}$$

17.1.30.4 Single Seepage Value

The Single Seepage Value method sets the seepage from the face of the dam equal to a scalar value.

SLOTS ADDED BY THIS METHOD:

☛ **SEEPAGE**

Type: Series
Units: FLOW
Description: Flow of water through the dam face
Information: Seepage is not included in the Outflow of the reservoir and will need to be linked separately if the water does in fact go downstream.
I/O: Optional Input
Links: Linkable

☛ **SINGLE SEEPAGE VALUE**

Type: Scalar
Units: FLOW
Description: seepage value to be applied to each timestep
Information:
I/O: Required Input
Links: Not Linkable

This method is executed at the beginning of the run. For each timestep from the initial timestep through the end of the run (plus post run dispatching timesteps too), if the **Seepage** is not input, the **Seepage** is set equal to the **Single Seepage Value**.

The method will issue an error if there is not a valid value in the **Single Seepage Value** slot. Note, this structure allows some flexibility. Seepage can be input/rules when necessary but will use the scalar value when not input.

17.1.30.5 Linked Seepage

This method is intended to be used when linking a Reservoir object with a Groundwater Storage object that uses the **Head Based Boundary Condition** method in the **Solution Type** category HERE (Objects.pdf, Section 14.1.1.3).

SLOTS ADDED BY THIS METHOD:

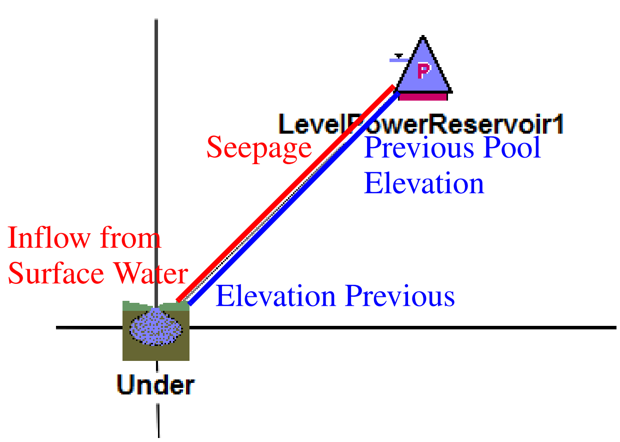
☛ **SEEPAGE**

Type: Series
Units: FLOW
Description: Flow of water out of the reservoir, often into groundwater
Information: A positive value is flow out of the reservoir.
I/O: Output only if linked to a Groundwater object (typical); otherwise required input
Links: Must be linked, typically to **Inflow from Surface Water** on Groundwater object

☛ **PREVIOUS POOL ELEVATION**

Type: Series
Units: LENGTH
Description: Pool Elevation at the end of the previous timestep
Information:
I/O: Output only
Links: Typically linked to the **Elevation Previous** slot on a Groundwater object

This method does not do any calculations; it just adds the appropriate slots. The Reservoir does provide the **Previous Pool Elevation** which is then linked to the Groundwater object **Elevation Previous**. The Groundwater object computes the **Inflow from Surface Water**, which is linked back to the **Seepage** slot on the reservoir. The **Seepage** is used in the Reservoir mass balance.



Click HERE (Objects.pdf, Section 14.1.1.3) for more information about how **Seepage** is calculated as **Inflow from Surface Water** on the linked Groundwater object.

17.1.31 Operating Levels

This category enables the user to specify operating levels for the reservoir. Operating levels serve as a normalizing metric for reservoir contents. This metric is used by reservoir-balancing algorithms to determine the relative “fullness” of reservoirs. On individual reservoirs, it also serves to identify the elevations that correspond to pool boundaries, such as the top of the conservation pool or the top of the flood pool.

17.1.31.1 None

This is the default method; no slots are instantiated and no calculations are performed.

17.1.31.2 Conservation Pools

This method allows the user to specify that there is a conservation pool for this reservoir.

Slots Specific to this Method

☛ **Operating Level Table**

Type: PeriodicSlot
Units: Time vs Length at Operating Levels
Description: table describing the seasonal variation of elevation in a reservoir at each of the user-designated operating levels.
Information: number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep.
I/O: Required Input
Links: Not Linkable

☛ **Operating Level**

Type: SeriesSlot
Units: None
Description: The computed operating level
Information: This slot is computed using the pool elevation and the Operating Level Table
I/O: Output only
Links: Not Linkable

☛ Operating Level Storage Table

Type: PeriodicSlot
Units: TIME VS VOLUME AT OPERATING LEVELS
Description: table describing the seasonal variation of storage in a reservoir at each of the user-designated operating levels.
Information: Each column represents the time-varying storage for a particular Operating Level. The integer value of the Operating Level is in the header of each column. This table is generated from the Operating Level Table and has the same number of rows and columns. The values in the table are storage values (looked up from the elevation volume table) whereas the values in the Operating Level Table are elevations. This slot is computed at run-time so it is read-only to the user. All changes should be made in the Operating Level Table.
I/O: Output Only
Links: Not Linkable

☛ Top of Conservation Pool

Type: ScalarSlot
Units: NONE
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the conservation pool.
Information:
I/O: Input
Links: Not Linkable

☛ Bottom of Conservation Pool

Type: ScalarSlot
Units: NONE
Description: Operating level (as defined in Operating Level Table) corresponding to the bottom of the conservation pool.
Information: Used by some conservation pool operations algorithms, along with the Top of Conservation Pool slot, to identify the volume in the conservation pool.
I/O: Input Only
Links: Not Linkable

☛ Conservation Pool Initial Empty Space

Type: SeriesSlot
Units: NONE
Description: The inflow required to fill the conservation pool at the beginning of timestep, based on the ending storage at the prior timestep, taking into account evaporation, precipitation, etc.

Information: This slot is computed at the beginning of the timestep; evaporation rates and other such factors that are not already defined at the beginning of timestep will not be taken into account in this computation. This slot is used by the water rights allocation solution algorithm to compute physical constraints and by storage accounts to compute allocation requests.
I/O: Output only
Links: Not Linkable

CONSERVATION POOL STORAGE

Type: Series
Units: VOLUME
Description: This is the computed volume of water in the conservation pool.
Information: This value is always non-negative.
I/O: Output only
Links: Not Linkable

CONSERVATION POOL FULL STORAGE

Type: Series
Units: VOLUME
Description: This is the possible volume of water that could be stored in the Conservation pool. It is computed as the Storage at the top of the conservation pool minus the storage at the bottom of the conservation pool
Information:
I/O: Output Only
Links: Not Linkable

CONSERVATION POOL STORAGE FRACTION

Type: Periodic Slot
Units: FRACTION
Description: The values in the periodic slot represent the percentage of the conservation pool storage at each level (column) in the Operating Level Table.
Information: It has identical dimension including dates and levels as the Operating Level Table. This table will be populated at beginning of run. The Operating Level Table will be its “source” slot.
I/O: Output only
Links: Not Linkable

At the beginning of run, the Conservation Pool Storage Fraction is populated as follows: For each date (row) and each level, n, (column), the equation to compute the fraction:

$$\text{Conservation Pool Storage Fraction}[date,n]=$$

$$\frac{\text{Op Level Storage Table}[date, n] - \text{Op Level Storage Table}[date, \text{Bottom of Cons Pool}]}{\text{Op Level Storage Table}[date, \text{Top of Cons Pool}] - \text{Op Level Storage Table}[date, \text{Bottom of Cons Pool}]}$$

Note, the Conservation Pool Storage Fraction is limited to be always between 0 and 1 (0% to 100%).

At the end of each dispatch method, the Operating Level series slot is computed by looking up the pool elevation and date on the Operating Level Table.

Next, **Conservation Pool Full Storage** is calculated as follows.

$$\text{Conservation Pool Full Storage}[t] = \text{Operating Level Storage Table}[t, \text{Top of Conservation Pool}] - \text{Operating Level Storage Table}[t, \text{ Bottom of Conservation Pool}]$$

The **Conservation Pool Storage** is computed as:

If the Operating Level is greater than the Top of the Conservation Pool,

$$\text{Conservation Pool Storage}[t] = \text{Full Conservation Pool Storage}[t]$$

else if the Operating Level is less than the Bottom of the Conservation Pool,

$$\text{Conservation Pool Storage}[t] = 0$$

else

$$\text{Conservation Pool Storage}[t] = \text{Storage}[t] - \text{Operating Level Storage Table}[t, \text{ Bottom of Conservation Pool}]$$

17.1.31.3 Conservation and Flood Pools

This method allows the user to specify that there is a conservation and a flood pool for this reservoir.

Slots Specific to this Method

This method is an extension of the Conservation Pool method, and selecting this method causes all the slots for Conservation Pool to become available, along with the following:

☛ Top of Flood Pool

Type: ScalarSlot
Units: No Units
Description: Operating level (as defined in Operating Level Table) corresponding to the top of flood pool.

Information:
I/O: Required Input
Links: Not Linkable

☛ Flood Pool Storage

Type: Series
Units: Volume
Description: This is the computed volume of water in the flood pool.
Information: This value is always non-negative.
I/O: Output Only
Links: Not Linkable

☛ Flood Pool Full Storage

Type: Series
Units: Volume
Description: This is the possible volume of water that could be stored in the Flood pool. It is computed as the Storage at the top of the flood pool minus the storage at the top of the conservation pool.
Information:
Links: Not Linkable

☛ Flood Pool Storage Fraction

Type: Periodic Slot
Units: Fraction
Description: The values in the periodic slot represent the percentage of the flood pool storage at each level (column) in the Operating Level Table.
Information: It has identical dimension including dates and levels as the Operating Level Table. This table will be populated at beginning of run. The Operating Level Table will be its “source” slot.
I/O: Output only
Links: Not Linkable

At the beginning of run, the Conservation Pool Storage Fraction is populated as described above. Then, the Flood Pool Storage Fraction is populated as follows: For each date (row) and each level, n, (column), the equation to compute the fraction is:

$$\text{Flood Pool Storage Fraction}[date,n] =$$

$$\frac{\text{Op Level Storage Table}[t, n] - \text{Op Level Storage Table}[t, \text{Top of Cons Pool}]}{\text{Op Level Storage Table}[t, \text{Top of Flood Pool}] - \text{Op Level Storage Table}[t, \text{Top of Cons Pool}]}$$

Note, the Flood Pool Storage Fraction is limited to be always greater than 0. But, it can be larger than 1 (100%). For levels above the flood pool, the percentage will be greater than 100%.

At the end of each dispatch method, the Operating Level series slot is computed by looking up the pool elevation and date on the Operating Level Table. Next, all slots associated with the Conservation Pool are computed and set as described above. Then, **Flood Pool Full Storage** is calculated as follows:

$$\text{Flood Pool Full Storage}[t] = \text{Operating Level Storage Table}[t, \text{Top of Flood Pool}] - \text{Operating Level Storage Table}[t, \text{Top of Conservation Pool}]$$

The **Flood Pool Storage** is computed as:

If the Operating Level is less than the Top of the Conservation Pool,

$$\text{Flood Pool Storage}[t] = 0$$

else

$$\text{Flood Pool Storage}[t] = \text{Storage}[t] - \text{Operating Level Storage Table}[t, \text{Top of Conservation Pool}]$$

Note, the **Flood Pool Storage** may be larger than the **Flood Pool Full Storage**. This indicates the reservoir is above the flood pool and is surcharging.

17.1.32 Conditional Operating Levels

This category provides methods that allows the user to use alternative operating level tables based on conditions in the run.

17.1.32.1 None

This is the default method; no slots are instantiated and no calculations are performed. The original Operating Level Table is used for all computations.

17.1.32.2 Sum Inflows over Interval

This method allows an alternative operating level table (i.e. a guide curve) to be used starting on a certain date if a certain combination of flows are high enough for a specified time range.

For example, if there has been a total of 200,000 acre-feet of total inflows into a specific reservoir during the months of March, April, and May, then on June 15th, the method would switch the reservoir operations to follow an alternative table. On October 15th, the reset date, the reservoir will once again use the original Operating Level Table.

Slots Specific to this Method

☛ **Operating Level 2 Table**

Type: Periodic
Units: Time vs Length at Operating Levels
Description: This is the alternative operating level table that is used when indicated by hydrologic conditions. This table describes the seasonal variation of elevation in a reservoir at each of the user-designated operating levels.
Information: Number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep. This table should have the same dimensions (rows and columns) as the Operating Level Table.
I/O: Required Input
Links: Not Linkable

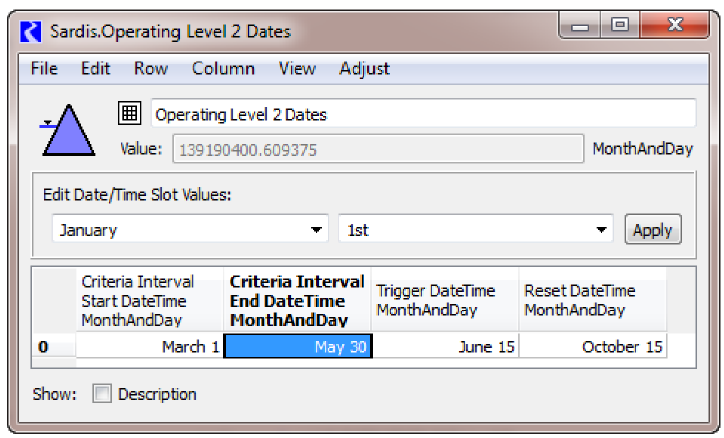
☛ **Operating Level 2 Trigger Volume**

Type: Scalar
Units: Volume

Description: The volume of inflows (**Inflow Sum** slot) between the criteria start and finish (specified on the **Operating Level 2 Dates** slot) that causes the reservoir to use the **Operating Level 2 Table** rather than the original **Operating Level Table.**
I/O: Required Input or an error will be issued at the start of run.
Links: NA

☛ Operating Level 2 Dates

Type: Table
Units: DateTime
Description: This table has 4 columns. The first and second columns are the “Criteria Interval Start DateTime” and “Criteria Interval End DateTime” between which the flow volume is summed and compared to the **Operating Level 2 Trigger Volume**. The third column is the “Trigger DateTime” at which the reservoir will check the conditions and possibly use the **Operating Level 2 Table** rather than the original **Operating Level Table.** The fourth column is the “Reset DateTime” at which the reservoir will use the **Operating Level Table** again.
Information: The units for this slot are DateTime which can be an absolute or partially specified datetime. The default user units are “MonthAndDay”. Partially specified datetimes are converted to fully specified datetimes using information from the current timestep to fill in the missing pieces of the partially specified timestep. Thus, if the datetime is partially specified, it must be able to evaluate to a timestep in the model or an error will be issued.



I/O: Required input or an error will be issued at the start of run.
Links: NA

☛ Operating Level Storage 2 Table

Type: Periodic
Units: Storage
Description: This represents the Storage associated with the elevations in the Operating Level 2 Table.

Information: This slot is created at the beginning of run. The Operating Level 2 Table is its "source" slot.
I/O: Output Only
Links: Not Linkable.

☛ Conservation Pool Storage Fraction 2

Type: Periodic Slot
Units: FRACTION
Description: The values in the periodic slot represent the percentage of the conservation pool storage at each level (column) in the Operating Level Storage 2 Table.
Information: It has identical dimension including dates and levels as the Operating Level Storage 2 Table. This table will be populated at beginning of run. The Operating Level 2 Table will be its "source" slot.
I/O: Output only
Links: Not Linkable

☛ Flood Pool Storage Fraction 2

Type: Periodic Slot
Units: FRACTION
Description: The values in the periodic slot represent the percentage of the flood pool storage at each level (column) in the Operating Level 2 Table.
Information: It has identical dimension including dates and levels as the Operating Level Storage 2 Table. This table will be populated at beginning of run. The Operating Level 2 Table will be its "source" slot.
I/O: Output only
Links: Not Linkable

Method Details:

This method category will be dependent on the selection of the **Conservation and Flood Pools** or **Conservation Pool** method in the **Operating Levels** method category.

At the beginning of the run, the Operating Level Storage 2 Table will be populated by looking up the elevation values in the Operating Level 2 Table on the Elevation Volume Table to get the storage associated with each level. Next the Conservation Pool Storage Fraction 2 and Flood Pool Storage Fraction 2 slots will be populated as follows:

For each date (row) and each level, n, (column), the equation to compute the fraction:

$$\text{Conservation Pool Storage Fraction 2}[date,n]=$$

$$\frac{\text{Op Level Storage 2 Table}[t, n] - \text{Op Level Storage 2 Table}[t, \text{Bottom of Cons Pool}]}{\text{Op Level Storage 2 Table}[t, \text{Top of Cons Pool}] - \text{Op Level Storage 2 Table}[t, \text{Bottom of Cons Pool}]}$$

Note, the **Conservation Pool Storage Fraction 2** is limited to be always between 0 and 1 (0% to 100%).

For each date (row) and each level, n, (column), the equation to compute the fraction:

$$\text{Flood Pool Storage Fraction}[date,n] = \frac{\text{Op Level Storage 2 Table}[t, n] - \text{Op Level Storage 2 Table}[t, \text{Top of Cons Pool}]}{\text{Op Level Storage 2 Table}[t, \text{Top of Flood Pool}] - \text{Op Level Storage 2 Table}[t, \text{Top of Cons Pool}]}$$

Note, the Flood Pool Storage Fraction 2 is not limited to be between 0 and 1 (0% to 100%). For levels above the flood pool, the percentage will be greater than 100%.

Then, at the beginning of each timestep, the Sum Inflows over Interval method will check to see if the controller is on the “Trigger DateTime”. If so, the Inflow Sum slot will be summed (as a volume) over the criteria interval. If the sum is equal to or greater than the Operating Level 2 Trigger Volume, the reservoir will use the Operating Level 2 Table and Operating Level Storage 2 Table in all computations (until reset).

If the current timestep is a “Reset DateTime”, then the reservoir will again use the original Operating Level Table(s).

If the current timestep is neither a “Trigger DateTime” or a “Reset DateTime”, then the reservoir will reference the table used on the previous timestep. That is, it will not modify the table used but continue to use whichever table is in effect.

17.1.33 Target Operation

The Target Operation category is used to enable algorithms which calculate different lumped mass balance algorithms required by Target Operations (see TARGET flag). None of the methods exist as separate functions, meaning that the target operation algorithms themselves must be used in conjunction with the other mass balance algorithms. It is important to note that either the Simple Target or Lagged Target method must be selected if a target operation is set on the Reservoir.

17.1.33.1 None

This is the default method in the Target Operation category. It performs no calculations. It may only be selected if a Target Operation is not performed on the object. There are no slots specifically associated with this method.

17.1.33.2 Simple Target

The Simple Target method distributes the required Inflow or Outflow evenly among all available timesteps. The total flow is divided among all the non-input timesteps included within the Target Operation to meet the Target. For most cases with a target operation, this is the desired behavior. There are no slots specifically associated with this method.

Note: Evaporation and precipitation are not included in the lumped mass balance of the target operation.

17.1.33.3 Lagged Target

The Lagged Target method should be used when there are lag times in the upstream reaches. When Inflows are known, this method solves for the same solution as the Simple Target method. When Outflows are known, this method distributes the required Inflow so a steady Outflow occurs from an upstream, non-integer timestep lagged, reservoir. To distribute Inflows, the Total Lag of Upstream Reaches and Outflow from Upstream Reservoir are used to calculate the required Inflows to the current Reservoir. These required Inflows must meet the Target and result in steady Outflow from the upstream Reservoir. The solution yields a perturbed Inflow at the first undetermined timestep of the Target Operation, followed by steady Inflows for the remainder of the undetermined target times. This solution removes the numerical instability which would be produced in the Outflow of the upstream reservoir using the Simple Target method.

Slots Specific to this Method

☛ **Outflow from Upstream Reservoir**

Type: Series
Units: FLOW

Description: outflow from the upstream reservoir
Information:
I/O: Required input
Links: Should be linked to the Outflow slot of the upstream reservoir.

☛ TOTAL LAG OF UPSTREAM REACHES

Type: Table
Units: TIME
Description: combined lag time of reaches between the current and the upstream reservoir
Information:
I/O: Required input
Links: Not linkable

17.1.34 Sediment

The Sediment category is used to enable algorithms which adjust reservoir Elevation Volume and possibly Elevation Area relationships in response to sediment inflow. See also the “Time Varying Elevation Area” method **HERE (Objects.pdf, Section 17.1.35.2)** for more information on methods that change the elevation area relationship.

17.1.34.1 None

The None method is the default for the Sediment category. No calculations are performed in this method. There are no slots specifically associated with this method.

17.1.34.2 CRSS Sediment

The **CRSS Sediment** method is designed based on sedimentation calculations performed by the US Bureau of Reclamation’s Colorado River Simulation System (CRSS) model. This function distributes reservoir sediment based on the “Empirical Area Reduction Method”. Simply put, sediment is distributed through an iterative process in which a total volume loss due to sedimentation is calculated based on an assumed top of sediment elevation.

Slots Specific to this Method

☛ **Elevation Area Table**

Type: Table
Units: LENGTH vs. AREA
Description: generated elevation area table for calculating sediment distribution
Information:
I/O: Output only
Links: Not linkable

☛ **Elevation Vol_Area Table Increment**

Type: Table
Units: LENGTH
Description: elevation increments for the generated Elevation Volume and Elevation Area Tables
Information: This table often needs more precise elevation increments than the sediment calculation tables.
I/O: Required input
Links: Not linkable

☛ **Initial Elevation Area Table**

Type: Table
Units: LENGTH vs. AREA

Description: initial elevation area table
Information: Provided for comparison with initial data
I/O: Output only
Links: Not linkable

☛ Initial Elevation Volume Table

Type: Table
Units: LENGTH vs. VOLUME
Description: initial elevation volume table
Information: provided for comparison with initial data
I/O: Output only
Links: Not linkable

☛ Sediment Distribution Coefficients

Type: Table
Units: NOUNITS
Description: parameters for empirical equation governing sediment distribution
Information:
I/O: Required input
Links: Not linkable

☛ Sediment Inflow

Type: Series
Units: VOLUME
Description: volume of sediment flowing into the reservoir at each timestep
Information:
I/O: Required input
Links: Not linkable

☛ User Input Elev Area Data

Type: Table
Units: LENGTH vs. AREA
Description: initial Elevation Area relationship
Information: These values are initial conditions for the first timestep of the simulation. The elevation increments will be used for all sedimentation calculations.
I/O: Required input
Links: Not linkable

This volume loss is recalculated (with a new top of sediment elevation) at each iteration, until the calculated volume loss is equal to the actual volume of sediment inflow (within a specified convergence). The total volume loss calculation consists of a somewhat complicated algorithm utilizing elevation/area and elevation/volume data for the reservoir and an empirical equation. The empirical equation uses user specified parameters which relate the portion of total area that is taken up by sediment to the Pool Elevation. The empirical equation basically gives the shape of the accumulated sediment. The empirical equation has a close relationship to the elevation volume and elevation area characteristics of a given reservoir. The elevation/area and elevation/volume data is stored in a polynomial coefficient table, which gets recalculated after each timestep. The actual Elevation Area, Elevation Volume tables used by **RiverWare**™ are adjusted at the end of the sedimentation code (but prior to the hydrologic simulation).

Caution should be exercised in creating input data for this method. The close relationship between the empirical area reduction equation and the shape of the reservoir (reflected in the User Input Elev Area Data) makes the method fairly sensitive to input data. When choosing empirical parameters for this method, physical characteristics of the given reservoir need to be considered. The Bureau of Reclamation currently considers 4 possible types of reservoirs, with each type having a corresponding set of empirical area reduction parameters. The reservoir type classification is based on the shape of the Reservoir, the manner in which the reservoir is to be operated, and the size of the sediment particles to be deposited in the reservoir. The main emphasis is on the shape. Tables are used to classify the reservoirs based on these characteristics. Once the type has been established, the parameter values for that type can also be taken from tables in the literature. An incorrect set of parameters for a given reservoir will lead to an inability to achieve convergence on the sediment distribution within this method.

17.1.34.3 Time Varying Elevation Volume

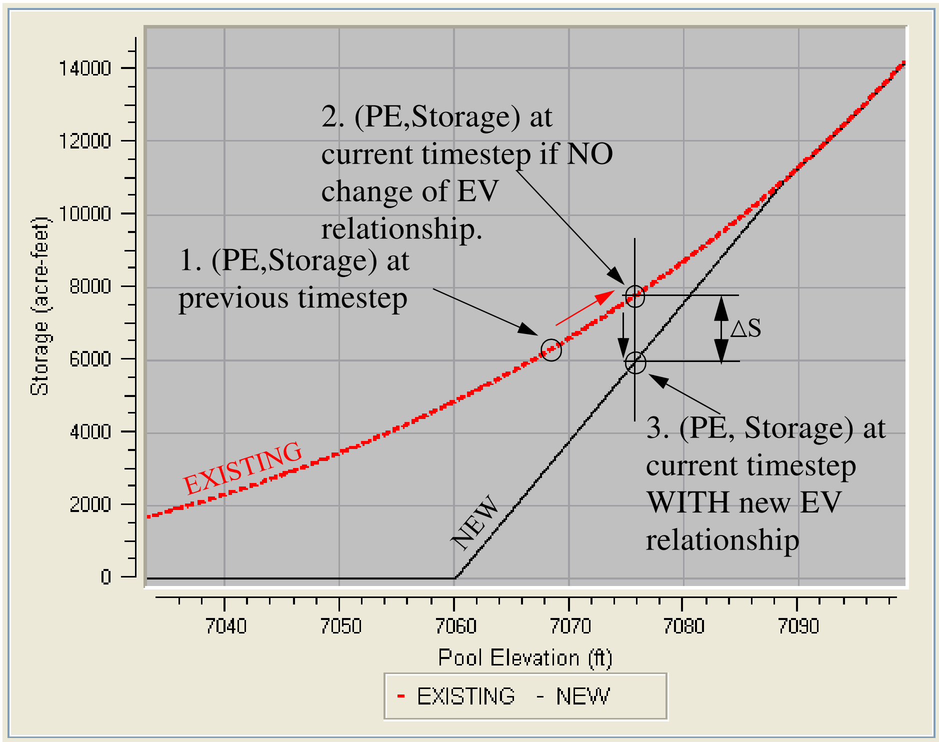
The **Time Varying Elevation Volume** method allows Elevation Volume (EV) relationships that change at specified times. The method is only available when the following default methods are selected in the following categories:

- Flood Control Release: only the None method is allowed
- Surcharge Release: only the None method is allowed
- Water Quality: None, Water Quality cannot be enabled

In the figure below, both the existing and new EV relationships are plotted. Shown is a graphical example of how the mass balance should be performed on the timestep the EV relationship changes. This process is described below.

1. This is the PE and Storage at the previous timestep. This is the starting condition.

2. On the current timestep, the Inflows and Outflows lead to a positive net inflow. This will increase the Pool Elevation. Using the existing table, a (PE, Storage) is calculated and shown as point 2.



3. The PE is used to interpolate a storage on the new EV table. The difference between storage at point 2 and point 3 is the loss of storage term. This means that the Pool Elevation is the same regardless of which table is used.

Slots Associated with this Method

☛ Elevation Volume Table Time Varying

Type: Table Slot

Units: Length and volume

Description: the tables that represent the Elevation Volume relationship at various times in the run.

Information: The number of columns in this table should be set to one plus the number of times the Elevation Volume relationship changes. The column headings contain the date corresponding to the change. When you add a column to this slot, it is given a date later then the last column. You can set the date from the Column -> Set Column Value menu. You can then type the date text or use the date time spinner to enter the appropriate date. The dialog will only let you enter dates that correspond to timesteps. (Care should be exercised when switching model timesteps) The column is then placed in the correct order compared to the other column labels. Each column should have an entry for each row in the Elevation column or an interpolation error may be issued during the run. The number of columns is equal to the number of changes plus one, i.e., if there is one change then there should be two columns. An example is shown below for a run that starts in 24:00 Jan 1, 1910 and the reservoir Elevation Volume changes three times. The column label of the first set of volumes must be on or before the initial timestep. The times on the column map are an instant in time.

I/O: Input Only
Links: Not Linkable

Pool Elevation ft	24:00 Jan 1, 1910 Storage acre-ft	24:00 Jan 1, 1935 Storage acre-ft	24:00 Jan 1, 1953 Storage acre-ft	24:00 Jan 1, 1970 Storage acre-ft
5,100	0	0	0	0
5,120	10	9	8	7
5,150	15	14	13	12
5,200	20	19	18	17
5,250	25	24	23	22

☛ Storage Adjustment from Elev Vol Table Change

Type: Series Slot
Units: VOLUME
Description: This is the volume of water that was lost to sedimentation
Information: The slot tracks the mass discontinuity that occurs when the Elevation Volume is changed because of a new reservoir Elevation Volume relationship. A positive number indicates storage was gained; a negative number indicates storage was lost.
I/O: Output Only
Links: Not Linkable

Method Details

Start of Run: The run proceeds as follows: at the start of the run, if the **Time Varying Elevation Volume** method is selected, a boolean variable isTimeVaryingElevVolume will be set to TRUE. At the same time, a pointer is set that specifies that all computations should use the **Elevation Volume Table Time Varying**. All computations involving the elevation volume relationship on the object use this pointer instead of the directly accessing the **Elevation Volume Table**. If the method is not selected, then the pointer is set to the **Elevation Volume Table**.

Note: Even with this method selected, the original Elevation Volume Table is still visible. Although it is not used when dispatching or other simulation or rulebased simulation, it is still a general slot that is used in optimization calculations and water quality.

Start of Each Dispatch Method: At the start of each dispatch method, if the isTimeVaryingElevVolume is true, then the method will determine which column ,"col", of the table to use. It compares the current timestep to the column headings on the table and determine which column to use. For example, if the current timestep is March 3, 1940 for the table example above, then it will set col = 2 (column numbering is zero based) for use in all remaining computations. If the current timestep exactly matches one of the column headings, then an additional variable, isElevVolModDate, is set to TRUE and col is set to the that column minus 1. That is, for this dispatch, the previous relationship will be used, but will be adjusted at the end of the method. Note, if the **Time Varying Elevation Volume** method is not selected, then isTimeVaryingElevVolume and isElevVolModDate will remain false and the column to use on the **Elevation Volume Table** is set to 1.

The dispatch method then proceeds as normal using the computed "col" in all references to the specified elevation volume table.

End of Each Dispatch Method: At the end of the dispatch method (the description applies first to solveMB_GivenInflowOutflow), once PE and Storage are known, if the isElevVolModDate is true, the method will lookup the PE on the **Elevation Volume Table Time Varying** but this time use col+1 and get S'. The new storage S' is the reduced storage using the new EV relationship. The difference, ∆S = S - S', and is set on the **Storage Adjustment from Elev Vol Table Change** slot. Then the Storage slot is set to S' and PreviousStorage[t+1] is set to S'.

The above procedure describes the solveMB_givenInflowOutflow. This same procedure is used for: solveMB_givenEnergyInflow and solveMB_givenInflowRelease. Once the new storage using the existing table is calculated, the new relationship can be used. If Pool Elevation (HW) is given (i.e. solveMB_givenOutflowHW), then the Pool Elevation is used to look up the storage using the existing and new tables. For the dispatch methods where Storage is known, the method will abort the run with an error that the **Time Varying Elevation Volume** method cannot be used on a timestep where the storage is given.

Limitations: This method changes fundamental information about the reservoir. As a result, there are certain operations that cannot be used with this method including:

- Target operations that span table modification dates
- Dispatching the reservoir with any of the "givenStorage" methods on a modification date. Non modification dates can use the "givenStorage" methods.
- Any of the following RPL functions: StorageToElevation, ElevationToStorage, and StorageToArea. If these functions are called on a reservoir with the **Time Varying Elevation Volume** selected, an error message will be posted. Instead use the "...AtDate" version of that function. I.e. use the StorageToElevationAtDate instead of the StorageToElevation function. Old models may need to be updated.

- There are many RPL functions like SolveOutflow, SolveStorage, GetMaxOutflowGivenInflow, etc that access the elevation volume relationship. These will access the correct table, but will always assume that the computation is being performed BEFORE any modifications to the relationship are made. That is, if you call the function and it is a modification timestep on the table, the function will use the previous column in all its computations. The relationship change is only considered at the end of the dispatch method, not in the RPL function.

17.1.35 Surface Area Modification

The **Surface Area Modification** category is dependent on having a valid evaporation method selected (i.e. any evaporation method except **None** or **MonthlyEvaporationCalcInAnnual**). The category will be added to all reservoirs and the default method in this category is **None** which performs no calculations and has no slots.

17.1.35.1 None

This is the default, no-action method.

17.1.35.2 Time Varying Elevation Area

The **Time Varying Elevation Area** method allows Elevation Area (EA) relationships that change at specified times. The **Time Varying Elevation Area** method is only available on Level Power and Storage reservoirs. Also, the method will only be available when the following default methods are selected in the following categories:

- Flood Control Release: only the None method is allowed
- Surcharge Release: only the None method is allowed
- Water Quality: None; Water Quality must be disabled

Slots Associated With This Method

☛ Elevation Area Table Time Varying

Type: Table Slot
Units: Length and Area
Description: the tables that represent the Elevation Area relationship at various times in the run.
Information: The number of columns in this table should be set to one plus the number of times the Elevation Area relationship changes. The column headings contain the date corresponding to the change. When you add a column to this slot, it is given a date later then the last column. You can set the date from the Column -> Set Column Value menu. You can then type the date text or use the date time spinner to enter the appropriate date. The dialog will only let you enter dates that correspond to timesteps. (Care should be exercised when switching model timesteps.) The column is then placed in the correct order compared to the other column labels. Each column should have an entry for each row in the Pool Elevation column or an interpolation error may be issued during the run. The number of columns is equal to the number of changes plus one, i.e., if there is one change then there should be two columns. An example is shown below for a run that starts in Jan 1, 1910 and the reservoir Elevation Area changes three times. The column label of the first column of

areas must be on or before the initial timestep. The times on the column map are an instant in time.

I/O: Input Only
Links: Not Linkable

Pool Elevation ft	24:00 Jan 1, 1910 Surface Area acre	24:00 Jan 1, 1935 Surface Area acre	24:00 Jan 1, 1953 Surface Area acre	24:00 Jan 1, 1970 Surface Area acre
5,100	0	0	0	0
5,120	10	9	8	7
5,150	15	14	13	12
5,200	20	19	18	17
5,250	25	24	23	22

☛ Surface Area Adjustment to Elev Area Table Change

Type: Series Slot
Units: Area
Description: The surface area lost to sedimentation that occurs when the reservoir is resurveyed and a new Elevation Area Table is implemented.
Information: A positive number indicates area was added. A negative number indicates surface area was lost.
I/O: Output Only
Links: Not Linkable

Method Details

The evaporation methods typically use the average Surface Area in the calculations. Because the average Surface Area may change when the new table becomes active, the computations assume that the Surface Area only changes at the end of a dispatch method. That is, on a modification timestep, the previous relationship is used in the evaporation and precipitation methods, then the Surface Area is modified at the end of the dispatch method. Thus, the modified surface area will be reflected in evap and precip computations at the NEXT timestep.

Note: Even with this method selected, the original Elevation Area Table is still visible. Although it is not used when dispatching or other simulation or rulebased simulation, it is still a general slot that is used in optimization calculations and water quality.

The run proceeds as follows:

Start of Run: At the start of the run , if the **Time Varying Elevation Area** method is selected, a boolean variable isTimeVaryingElevArea will be set to TRUE. At the same time, a pointer is set that specifies to use the **Elevation Area Table Time Varying** table. All other computations now reference this pointer instead of the **Elevation Area Table**. If the method is not selected, then the pointer is set to the **Elevation Area Table**.

In each Evap/Precip Method: At the start of each method, if the isTimeVaryingElevArea is true, then the method will determine which column ,"col", of the table to use. It will compare the current timestep to the column headings on the table and determine which column to use. For example, if the current timestep is March 3, 1940 for the table example above, then it will set col = 2 (column numbering is zero based) for use in all remaining computations.

If the current timestep exactly matches one of the column headings, col is set to that column minus 1. That is, for this method, the previous relationship will be used. Also, the variable isElevAreaModDate, is set to TRUE. The computation proceeds as normal. That is, the estimate for surface area uses the original relationship, not the modified relationship.

End of Dispatch method: At the end of the dispatch method, if the isElevAreaModDate is true, the method will lookup the PE on the **Elevation Area Table Time Varying** but this time use col+1 and get SA'. The new surface area SA' is the modified surface area using the new EA relationship. The difference, ΔSA = SA - SA', is set on the **Surface Area Adjustment to Elev Area Table Change** slot. Then the Surface Area slot is set to SA'. This does not affect the evaporation or precipitation computation on this timestep.The new relationship will be used on the next timestep though.

Note, if the **Time Varying Elevation Area** is not selected, then isTimeVaryingElevArea and isElevAreaModDate will remain false and the column to use on the **Elevation Area Table** is set to 1.

Limitations: This method changes fundamental information about the reservoir. As a result, there are certain operations that cannot be used with this method including:

- Any of the following RPL functions: StorageToArea and ElevationToArea. If these functions are called on a reservoir with the **Time Varying Elevation Area** method selected, an error message will be posted. Instead use the "...AtDate" version of that function. I.e. use the ElevationToAreaAtDate instead of the ElevationToArea function.
- There are many RPL functions like SolveOutflow, SolveStorage, GetMaxOutflowGivenInflow, etc that access the elevation area relationship. These will access the correct table, but will always assume that the computation is being performed BEFORE any modifications to the relationship are made. That is, if you call the function and it is a modification timestep on the table, the function will use the previous column in all its computations. The relationship change is only considered at the end of the dispatch method, not in the RPL function. The new relationship will be used on the next timestep though.

17.1.36 Additional Hydropower Release

The Additional Hydropower Release category is used to calculate the additional release necessary to meet a hydropower demand.

The method is dependent on having the Phase Balancing or Operating Level Balancing flood control method selected. In addition, this category is dependent on having the Peak Power Equation with Off Peak Spill power calculation method selected. There are two methods: the default, no-action "None" method and the "Meet Hydropower Load" method.

Use of this method for USACE-SWD is described HERE (USACE_SWD.pdf, Section 3.9).

17.1.36.1 None

This is the default method, there are no calculations or slots associated with this method.

17.1.36.2 Meet Hydropower Load

The Meet Hydropower Load method is used by the COE- Southwestern Division to calculate the additional hydropower release necessary to meet a hydropower load. This method is executed from the HydropowerRelease() predefined rule function.

The following slots are associated with the method:

☛ **ADDITIONAL HYDROPOWER RELEASE**

Type: Series Slot
Units: FLOW
Description: The additional power release limited to prevent additional downstream flooding
Information: The value in this slot IS limited to prevent additional downstream flooding
I/O: output only
Links: Not linkable

☛ **PROPOSED ADDITIONAL HYDROPOWER RELEASE**

Type: Series Slot
Units: FLOW
Description: The additional power release to meet a load
Information: The value in this slot IS NOT limited to prevent additional downstream flooding. It is temporary only, the values are only set at runtime but not saved in the model file.
I/O: output only
Links: Not linkable

This method determines the additional outflow required to meet the load; or if the load cannot be met, to maximize the power production. The function determines the outflow to be the minimum of:

1. The Outflow calculated to meet the given Load. The method uses the user selected methods on the reservoir, i.e. it calls the Get Peak Power Equation with Off Peak Spill method and the selected tailwater method.

2. The Outflow calculated such that the Pool Elevation would exactly equal the Minimum Power Pool Elevation.

3. The Outflow calculated such that the Pool Elevation exactly equals the previous Pool Elevation minus the Maximum Power Pool Drawdown slot.

4. The Outflow that generate the maximum possible Energy. This Energy is produced by running the turbines at max release (generator capacity) for the full timestep.

The method sets the Proposed Additional Hydropower Release slot and also returns the additional flow to the calling rule function. The rule function HydropowerRelease() then performs a check to ensure that no additional downstream flooding will occur. The rule then sets the Outflow and Additional Hydropower Release on the power reservoir.

For more information on the HydropowerRelease function, click **HERE (RPLPredefinedFunctions.pdf, Section 96)**.

17.1.37 Load Calculation

When the “Meet Hydropower Load” method is selected, a category called “Load Calculation” becomes visible. In this category there are seven methods: None, Input Load, Annual Load, Monthly Load, Periodic Load, Seasonal Load and Seasonal Load Time. Most of the selected method in this category are executed at the beginning of each timestep so that the method is only called once per timestep but the information on the state of the system is available (such as previous Pool Elevation) to the method. Input Load and Seasonal Load time are not called at the beginning of the timestep as Input Load doesn’t have any computations and Seasonal Load Time is executed as described below. Below is a description of each method.

17.1.37.1 None

This is the default method and has no slots associated with it. If this method is selected, an error is issued saying “Cannot find a selected method in the Load Calculation category.”

17.1.37.2 Input Load

If this method is selected, the Load must be input or specified with a rule. If not valid, the run will abort with an error.

Slots associated with the Input Load method:

☛ **LOAD**

Type: Series Slot
Units: ENERGY
Description: The desired energy that the power reservoir should produce
Information:
I/O: Required Input or set by a rule
Links: Not linkable

☛ **DUMP ENERGY**

Type: Series Slot
Units: ENERGY
Description: The exceed energy produced during the timestep.
Information: Calculated as Max(0, Energy - Load) at the end of the “Peak Power Equation with Off Peak Spill” power method
I/O: Output only
Links: Not Linkable

☛ **THERMAL PURCHASE**

Type: Series Slot

Units: ENERGY
Description: Energy shortage for the given timestep. It is the amount of additional energy that would be required to fully meet the Load
Information: Calculated as max (0, Load - Energy) at the end of the “Peak Power Equation with Off Peak Spill” power method
I/O: Output only
Links: Not Linkable

☛ OPERATION FACTOR

Type: Series Slot
Units: DECIMAL
Description: The fraction of the timestep at which the turbines are running
Information: Calculated as Peak Time / Timestep Length. This is done at the end of the “Peak Power Equation with Off Peak Spill” power method
I/O: Output Only
Links: Not Linkable

17.1.37.3 Annual Load

For the Annual Load method, the user provides the load values for each year of the run and a set of weekday and weekend monthly factors used to disaggregate the annual load to daily values.

Slots associated with the Annual Load method:

☛ ANNUAL LOAD

Type: Series Slot
Units: ENERGY
Description: The load for each year of the run.
Information: The user must set the timestep for this series slot to be yearly. See the example in the table below. It is important to note that because this slot is annual, it is most likely different than the run timestep. As a result, if the user desires to “syncronize objects”, the user must select the toggle in the synchronization control to “Exclude Slots with Different Timestep from Run.” This will prevent changing the timestep of this slot when other slots are synchronized.
I/O: Input only
Links: Not linkable

☛ LOAD

Type: Series Slot
Units: ENERGY
Description: The desired energy that the power reservoir should produce

Information: The load value is either a user input or set by a method in the Load Calculation category
I/O: Input or set by the method
Links: Not linkable

LOAD FACTOR

Type: Periodic Slot
Units: NONE
Description: The factors used to disaggregate the yearly load to daily load
Information: The periodic slot contains a value for each month of the year. The first data column is the multiplying factor to disaggregate the annual load to a daily value for weekdays. The second data column contains the disaggregating factors for weekends. See the example in the table below.
I/O: Input only
Links: Not Linkable

Sample slots for the Annual Load Method

Annual Load	
Month	Load (KWH)
24:00 Dec 31, 1954	10,000
24:00 Dec 31, 1955	11,000
24:00 Dec 31, 1956	12,000
...	...

Load Factor		
Date	Weekday Load Factor	Weekend Load Factor
January 1	.041	0
February 1	.041	0
March 1	.083	.041
...	...	...

DUMP ENERGY

Type: Series Slot
Units: ENERGY
Description: The exceed energy produced during the timestep.
Information: Calculated as Max(0, Energy - Load) at the end of the “Peak Power Equation with Off Peak Spill” power method
I/O: Output only
Links: Not Linkable

THERMAL PURCHASE

Type: Series Slot
Units: ENERGY
Description: Energy shortage for the given timestep. It is the amount of additional energy that would be required to fully meet the Load

Information: Calculated as max (0, Load - Energy) at the end of the “Peak Power Equation with Off Peak Spill” power method
I/O: Output only
Links: Not Linkable

☛ **OPERATION FACTOR**
Type: Series Slot
Units: DECIMAL
Description: The fraction of the timestep at which the turbines are running
Information: Calculated as Peak Time / Timestep Length. This is done at the end of the “Peak Power Equation with Off Peak Spill” power method
I/O: Output Only
Links: Not Linkable

17.1.37.4 Monthly Load

For the Monthly Load method, the user provides the load values for each month of the run and a set of weekday and weekend monthly factors used to disaggregate the monthly values to daily values. The monthly load value is multiplied by the either the weekend or weekday factor and then divided by the number of weekdays or weekend days in the given month as applicable.

Slots associated with the Monthly Load method:

☛ **LOAD**
Type: Series Slot
Units: ENERGY
Description: The desired energy that the power reservoir should produce
Information: The load value is either a user input or set by a method in the Load Calculation category
I/O: Input or set by the method
Links: Not linkable

☛ **LOAD FACTOR**
Type: Periodic Slot
Units: NONE
Description: The factors used to disaggregate the monthly load to daily load
Information: The periodic slot contains a value for each month of the year. The first data column is the multiplying factor to disaggregate the monthly load to a daily value for weekdays. The second data column contains the disaggregating factors for weekends. See the example below.
I/O: Input only
Links: Not Linkable

Monthly Load

Type: Series Slot
Units: ENERGY
Description: The load for each month of the run.
Information: The user must set the timestep for this series slot to be monthly. See the example below. It is important to note that because this slot is monthly, it is most likely different than the run timestep. As a result, if the user desires to "synchronize objects", the user must select the toggle in the synchronization control to "Exclude Slots with Different Timestep from Run." This will prevent changing the timestep of this slot when other slots are synchronized.
I/O: Input only
Links: Not linkable

Sample slots for the Monthly Load Method

Monthly Load	
Month	Load (GWH)
24:00 Jan 31, 1940	1.71
24:00 Feb 28, 1940	1.56
24:00 Mar 31, 1940	1.72
...	...

Load Factor		
Date	Weekday Load Factor	Weekend Load Factor
January 1	1	0
February 1	1	0
March 1	.9	.1
...	...	...

Dump Energy

Type: Series Slot
Units: ENERGY
Description: The exceed energy produced during the timestep.
Information: Calculated as Max(0, Energy - Load) at the end of the "Peak Power Equation with Off Peak Spill" power method
I/O: Output only
Links: Not Linkable

Thermal Purchase

Type: Series Slot
Units: ENERGY
Description: Energy shortage for the given timestep. It is the amount of additional energy that would be required to fully meet the Load
Information: Calculated as max (0, Load - Energy) at the end of the "Peak Power Equation with Off Peak Spill" power method
I/O: Output only
Links: Not Linkable

☛ **Operation Factor**

Type: Series Slot
Units: DECIMAL
Description: The fraction of the timestep at which the turbines are running
Information: Calculated as Peak Time / Timestep Length. This is done at the end of the "Peak Power Equation with Off Peak Spill" power method
I/O: Output Only
Links: Not Linkable

17.1.37.5 Periodic Load

For the Periodic Load method, the user provides a periodic slot that contains the load for both weekday and weekend days. During the run, the load is looked up from this slot.

Slots associated with the Periodic Load method:

☛ **Load**

Type: Series Slot
Units: ENERGY
Description: The desired energy that the power reservoir should produce
Information: The load value is either a user input or set by a method in the Load Calculation category
I/O: Input or set by the method
Links: Not linkable

☛ **Periodic Load**

Type: Periodic Slot
Units: TIME VS ENERGY AND ENERGY
Description: A periodic slot containing the load in units of Energy to be used for the run.
Information: The slot consists of two columns, one column for the weekday values, one for the weekend values. See an example in the table below.
I/O: Input only
Links: Not linkable

Sample slots for the Periodic Load Method

Date	Weekday Load (KWH)	Weekend Load (KWH)
January 1	100	20

Sample slots for the Periodic Load Method

Date	Weekday Load (KWH)	Weekend Load (KWH)
February 1	120	30
March 1	120	20

☛ Dump Energy

Type: Series Slot
Units: Energy
Description: The exceed energy produced during the timestep.
Information: Calculated as Max(0, Energy - Load) at the end of the “Peak Power Equation with Off Peak Spill” power method
I/O: Output only
Links: Not Linkable

☛ Thermal Purchase

Type: Series Slot
Units: Energy
Description: Energy shortage for the given timestep. It is the amount of additional energy that would be required to fully meet the Load
Information: Calculated as max (0, Load - Energy) at the end of the “Peak Power Equation with Off Peak Spill” power method
I/O: Output only
Links: Not Linkable

☛ Operation Factor

Type: Series Slot
Units: Decimal
Description: The fraction of the timestep at which the turbines are running
Information: Calculated as Peak Time / Timestep Length. This is done at the end of the “Peak Power Equation with Off Peak Spill” power method
I/O: Output Only
Links: Not Linkable

The method looks up the current date in the date column of the table and sets the value from the appropriate column on the Load slot.

17.1.37.6 Seasonal Load

For the Seasonal Load method, the user provides two 3-dimensional periodic slots that contain the load for weekdays and weekends. During the run, the Load is looked up from the appropriate slot.

Slots associated with the Seasonal Load method:

☛ LOAD

Type: Series Slot
Units: ENERGY
Description: The desired energy that the power reservoir should produce
Information: The load value is either a user input or set by a method in the Load Calculation category
I/O: Input or set by the method
Links: Not linkable

☛ SEASONAL WEEKDAY LOAD

Type: Periodic Slot
Units: TIME VS ENERGY VS LENGTH
Description: A periodic slot containing the date, column headings containing the energy load, and values in the table representing pool elevations
Information: The first column contains the date. The remaining columns represent the load in increasing order. The values in the table represent the pool elevations. See the example in the table below.
I/O: Input only
Links: Not linkable

☛ SEASONAL WEEKEND LOAD

Type: Periodic Slot
Units: TIME VS ENERGY VS LENGTH
Description: A periodic slot containing the date, column headings containing the energy load, and values in the table representing pool elevations
Information: The first column contains the date. The remaining columns represent the load in increasing order. The values in the table represent the pool elevations.
I/O: Input only

Links: Not linkable

Sample slots for the Seasonal Load Method

	Load [MWH]			
Date	**100**	**200**	**300**	**400**
January 1	5550 [ft]	5558	5565	5590
February 1	5550	5558	5565	5590
March 1	5550	5552	5560	5580
...	...	...	...	...

☛ Dump Energy

Type: Series Slot
Units: ENERGY
Description: The exceed energy produced during the timestep.
Information: Calculated as Max(0, Energy - Load) at the end of the "Peak Power Equation with Off Peak Spill" power method
I/O: Output only
Links: Not Linkable

☛ Thermal Purchase

Type: Series Slot
Units: ENERGY
Description: Energy shortage for the given timestep. It is the amount of additional energy that would be required to fully meet the Load
Information: Calculated as max (0, Load - Energy) at the end of the "Peak Power Equation with Off Peak Spill" power method power method
I/O: Output only
Links: Not Linkable

☛ Operation Factor

Type: Series Slot
Units: DECIMAL
Description: The fraction of the timestep at which the turbines are running
Information: Calculated as Peak Time / Timestep Length. This is done at the end of the "Peak Power Equation with Off Peak Spill" power method
I/O: Output Only
Links: Not Linkable

When this method is executed, first the appropriate periodic slot is selected depending on the type of day. Next, the day of year is looked up in the first column and the previous pool elevation is looked up in the column header. The method then uses the value in the table for the energy. The load is then written to the Load slot. As usual with periodic slots, the time range, number of columns, and interpolation/lookup configuration can be changed.

17.1.37.7 Seasonal Load Time

For the Seasonal Load Time method, the user provides two 3-dimensional periodic slots that contain the fraction of a timestep that the turbines should operate for weekdays and weekends.

Slots associated with the Seasonal Load Time method:

☛ Load

Type: Series Slot
Units: ENERGY
Description: The desired energy that the power reservoir should produce
Information: For this method, the Load is set by a rule using the HydropowerRelease predefined function **HERE (RPLPredefinedFunctions.pdf, Section 96)**.
I/O: Output only - Set by a rule. It cannot be input or it may cause the HydropowerRelease rule to fail giving unexpected results.
Links: Not linkable

☛ Load Time

Type: Series Slot
Units: TIME
Description: The desired time that the reservoir should operate at full turbine capacity
Information: The Load Time is the appropriate Seasonal Weekday (or Weekend) Load Fraction value multiplied by the Length. This is done of the timestep.
I/O: Output only
Links: Not linkable

☛ Seasonal Weekday Load Fraction

Type: Periodic Slot
Units: TIME VS FRACTION VS LENGTH
Description: A periodic slot containing the date, column headings containing the fraction of timestep, and values in the table representing pool elevations
Information: The first column contains the date. The remaining columns represent the fraction of timestep in increasing order. The values in the table represent the pool elevations. The column headings can be configured to be either decimal (0-1) or percentages (0-100%). See the example in table below.
I/O: Input only

Links: Not linkable

☛ Seasonal Weekend Load Fraction

Type: Periodic Slot
Units: TIME VS FRACTION VS LENGTH
Description: A periodic slot containing the date, column headings containing the fraction of timestep, and values in the table representing pool elevations
Information: The first column contains the date. The remaining columns represent the fraction of timestep in increasing order. The values in the table represent the pool elevations. The column headings can be configured to be either decimal (0-1) or percentages (0-100%). See the example in the table below.
I/O: Input only
Links: Not linkable

Sample slot for the Seasonal Load Time Method

	Fraction of Timestep [%]			
Date	**10**	**20**	**30**	**40**
January 1	5550 [ft]	5558	5565	5590
February 1	5550	5558	5565	5590
March 1	5550	5552	5560	5580
...	...	...	...	...

☛ Dump Energy

Type: Series Slot
Units: ENERGY
Description: The exceed energy produced during the timestep.
Information: Calculated as Max(0, Energy - Load). If this method is selected, Load is not known until set by the rule calling the HydropowerRelease function. Therefore, for this method only, Dump Energy is calculated at the end of the timestep. It is not available to be accessed by other rules at the current timestep.
I/O: Output only
Links: Not Linkable

☛ Thermal Purchase

Type: Series Slot
Units: ENERGY
Description: Energy shortage for the given timestep. It is the amount of additional energy that would be required to fully meet the Load

Information: Calculated as max (0, Load - Energy). If this method is selected, Load is not known until set by the rule calling the HydropowerRelease function. Therefore, for this method only, Thermal Purchase is calculated at the end of the timestep. It is not available to be accessed by other rules at the current timestep.
I/O: Output only
Links: Not Linkable

☛ Operation Factor

Type: Series Slot
Units: Decimal
Description: The fraction of the timestep at which the turbines are running
Information: Calculated as Peak Time / Timestep Length. This is done at the end of the "Peak Power Equation with Off Peak Spill" power method
I/O: Output Only
Links: Not Linkable

When this method is executed, first the appropriate periodic slot is selected depending on the day of week. Next, the timestep is looked up in the first column and the previous pool elevation is looked up in the table. The method then uses the value in the column heading for the fraction of timestep. The Load Time is then calculated as the fraction of timestep times the timestep length. As usual with periodic slots, the time range, number of columns, and interpolation/lookup configuration can be changed.

In this method, the Load value is determined by the method but is not set until the rule calling the Hydropower Release function completes execution. This is different than the other methods in the Load Calculation category. A special function call determines the energy produced if the reservoir is run at generator capacity for the Load Time. This energy is equivalent to the Load and is used in all subsequent calculations where Load is required.

17.1.38 Startup

This category depends on selecting the Unit Power Table method, and describes how the monetary cost associated with starting up or shutting down a unit (turbine) will be modeled. There are two methods in this category, one which does not model these costs (effectively assigning them a value of 0) and one which uses a table describing the combined costs for starting up and shutting down a unit.

17.1.38.1 No Method (default)

This is the default, do-nothing method.

17.1.38.2 Unit Lumped Cost Method

For each Unit, this method lumps the cost of startup and shutdown into one value.

SLOTS ADDED BY THIS METHOD

Note, many of these slots have column or row dimensions based on the number of units. The rows/columns of these slots are expanded at the beginning of the run to match the value in the Number of Units slot. When first configuring this method, the user must enter the Number of Units, then run the model (stepping through 1 timestep is enough) to grow the slots to the right dimensions.

The following slots are instantiated when this method is selected:

☛ **UNIT STARTUP COST TABLE**

Type: TableSlot
Units: VALUE ($)
Description: This table will indicate the cost of startup/shutdown of each unit.
Information: There will be one column for each unit and one row that represents the cost of startup/shutdown.
I/O: Required input
Links: NA

☛ **UNIT STARTUP COST**

Type: AggSeriesSlot
Units: VALUE ($)
Description: There is one column for each unit indicating the cost of startup/shutdown.
Information: In simulation, the value of Unit Startup Cost for each unit is the Unit Startup[u] * Unit Startup Cost Table [u].
I/O: Output only
Links: NA

☛ Unit Startup

Type: AggSeriesSlot
Units: No Units
Description: A value of 1 indicates that the unit starts up at that date; otherwise the value is 0, indicating that the unit does not start up at that date.
Information: There is one column for each unit.
I/O: Output only
Links: NA

☛ Unit Shutdown

Type: AggSeriesSlot
Units: No Units
Description: A value of 1 indicates that the unit shuts down at that date; otherwise the value is 0, indicating that the unit does not shut down at that date.
Information: There is one column for each unit.
I/O: Output only
Links: NA

☛ Number Of Units Startup

Type: SeriesSlot
Units: No Units
Description: The number of units which start up at a given date. This value is the sum over the columns of Unit Startup.
Information:
I/O: Output only
Links: NA

☛ Number Of Units Shutdown

Type: SeriesSlot
Units: No Units
Description: The number of units which shut down at a given date. This value is the sum over the columns of Unit Shutdown.
Information:
I/O: Output only
Links: NA

☛ Plant Startup Cost

Type: SeriesSlot
Units: Value ($)
Description: The total startup cost for the plant. This value is the sum over the columns of the Unit Startup Cost.
Information:
I/O: Output only

Links: Linkable, typically to a Thermal object's System Startup slot.

Method Details

In Simulation, if the Unit Lumped Cost method is selected, startup and shutdown will be summarized as follows:

- Calculate Unit Startup[t,u] = max(Unit Is Generating[t,u] - Unit Is Generating[t-1,u], 0)
- Calculate Unit Shutdown[t,u] = max(Unit Is Generating[t-1,u] - Unit Is Generating[t,u], 0)
- Calculate Number Of Units Startup[t] = sum(Unit Startup[t])
- Calculate Number Of Unit Shutdown[t] = sum(Unit Shutdown[t])

Note, if the previous Unit Is Generating is not known, it is assumed that the unit is neither starting up or shutting down; Unit Startup and Unit Shutdown are set to zero. This may happens on the start timestep when the previous value is not known. Also, if the current Unit Is Generating is not valid, the method is exited without performing any computations or setting any slots.

This method will calculate the cost associated with startup/shutdown for each unit and the plant:

- Unit Startup Cost[t,u] = Unit Startup[t,u] * Unit Startup Cost Table[u]
- Plant Startup Cost[t] = Σ (Unit Startup Cost[t])

17.1.39 Head Loss

This category depends on the Unit Power Table method and contains methods for modeling additional head loss that occurs. This head loss may come from the configuration of the penstocks for bringing water to the turbines.

17.1.39.1 No Method (default)

In this method, there is no additional head loss to be used in the power calculation. In terms of penstock head loss, this method should be selected if the penstocks for the units are independent and the penstock losses are typically incorporated in the power data. Thus the power data is specified in terms of operating head.

17.1.39.2 Shared Penstock Head Loss method

In this method, there is additional head loss that results because units share a common penstock. The operating head losses in the penstock depend on the total turbine release and are shared for all units. The net head is calculated by subtracting penstock losses from the operating head. The unit data and power must be specified in terms of unit Net Heads instead of Operating Head.

SLOTS ADDED BY THIS METHOD

The following slots are instantiated when this method is selected:

☛ **SHARED PENSTOCK HEAD LOSS TABLE**

Type: TableSlot
Units: FLOW VS LENGTH
Description: This table shows head losses in a shared penstock as a function of total turbine release.
Information: The table has two columns: Turbine Release and Shared Penstock Loss.
I/O: Required Input
Links: NA

METHOD DETAILS

In simulation, when either the Unit Power Table or Unit Power Table Release method is called, it has an estimate of operating head calculated as average PE minus average tailwater elevation. Within these methods, if the Shared Penstock Head Loss method is selected, the code will look up an estimate of total Turbine Release and compute the additional head loss and subtract it from the existing operating head before use in any other equation. An iterative solution is needed if Turbine Release is not known, i.e. if this method is called from the Unit Power Table Release method.

17.1.40 Cavitation

This category depends on selecting the Unit Power Table method and contains methods for dealing with the problem of cavitation on turbines. Cavitation is the sudden formation and collapse of low-pressure bubbles in liquids by means of mechanical forces and this process can cause damage to turbines under certain operating conditions.

17.1.40.1 No Method (default)

This is the default, do-nothing method.

17.1.40.2 Unit Head and Tailwater Based Regions

This method allows the user to specify the regions of operation in which cavitation does NOT occurs, so that these regions can be avoided. These regions can be dependent on both operating head and tailwater.

Slots added by this method

Note, many of these slots have column or row dimensions based on the number of units. The rows/columns of these slots are expanded at the beginning of the run to match the value in the Number of Units slot. When first configuring this method, the user must enter the Number of Units, then run the model (stepping through 1 timestep is enough) to grow the slots to the right dimensions.

The following slots are instantiated when this method is selected:

☛ **Unit Power Cavitation Table**

Type: TableSlot
Units: Length, Length, Power, and Power
Description: This table represents the region of operation that does not cause cavitations.
Information: The table will have one block per unit and four columns per block: head, tailwater, and minimum power to prevent cavitation, and maximum power to prevent cavitation. Interpolation of this table will be used to determine the feasibility for each flow - head combinations in the Unit Power Table. Some combinations may not be feasible at any tailwater, and these combinations should not be used in optimization or simulation. Others will have a minimum tailwater level for feasibility. For some units, tailwater may not affect cavitation. In these cases two rows should be used for each head: one with minimum tailwater and one with maximum tailwater.
I/O: Required Input
Links: NA

☛ **Unit Cavitation Optimization Tolerances**

Type: Table

Unit 1				Unit 2			
Tailwater (ft)	Operating Head (ft)	Min. Non-Cav. Power (MW)	Max. non-Cav. Power (MW)	Tailwater (ft)	Operating Head (ft)	Min. Non-Cav. Power (MW)	Max. non-Cav. Power (MW)
2067	100	1	12	2067	100	2	10
2067	200	1.1	12.5	2067	200	3	11
2067	300	1.4	12.6	2067	300	3.5	12
2116	100	2	12	2116	100	2	10
2116	200	2.1	12.3	2116	200	3	11
2116	300	2.5	12.8	2116	300	3.5	12

Units: FRACTION, FRACTION
Description: Tolerance used to adjust the cavitation region in Optimization.
Information: This slot is used to “shrink” the cavitation region in Optimization to avoid the possibility of optimal solutions that when run in the Rulebased Simulation, just barely dip into a cavitation zone. For example, if 0.01 is specfied, this translates to giving the optimization a 1% cushion to avoid the cavitation zone. There is one row for each unit.
I/O: Optional Input
Links: NA

METHOD DETAILS

In Simulation, the dispatch method will execute this method once head, tailwater elevation, and power are computed for each unit. This method will determine if the computed head, tailwater elevation, and power fall outside of the minimum and maximum power to prevent cavitation regions. If so, the method will issue an error but not stop the run. If the method is called and there is no valid Unit Power, then the method is exited without performing any computations.

17.1.41 Avoidance Zones

This category depends on selecting the Unit Power Table method and contains methods for modeling the existence of undesirable regions of operation for turbines. There are two methods in this category, one which does not model avoidance zones at all, and one which

17.1.41.1 No Method (default)

This the default, do-nothing method; avoidance zones are not considered.

17.1.41.2 Unit Head Based Avoidance Zones

This method allows the user to specify a table that defines the conditions in which the turbines should not be operated.

Slots added by this method

Note, many of these slots have column or row dimensions based on the number of units. The rows/columns of these slots are expanded at the beginning of the run to match the value in the Number of Units slot. When first configuring this method, the user must enter the Number of Units, then run the model (stepping through 1 timestep is enough) to grow the slots to the right dimensions.

The following slots are instantiated when this method is selected:

☛ **Unit Avoidance Zones Table**

Type: TableSlot
Units: Length, Power, Power
Description: This table represents zones in the Unit Power Table that should be avoided for operations.
Information: The avoidance zone table has one block per unit and three columns per block: head, power at bottom of the avoidance zone, power at the top of the avoidance zone. This table effectively removes regions from the Unit Power Table. The regions removed might have to be interpolated between points in the table. Heads that appear in this table must appear in the Unit Power Table as well.
I/O: Required Input
Links: NA

Unit 1			Unit 2		
Operating Head (ft)	Min. Power at Zone Bottom (MW)	Max. Power at Zone Top (MW)	Operating Head (ft)	Min. Power at Zone Bottom (MW)	Max. Power at Zone Top (MW)
100	4.5	6	100	8	9
200	5	6.4	200	9	9.5
300	6	7	300	9.2	9.9

METHOD DETAILS

In Simulation, the dispatch method will execute this method once head and power are computed for each unit. This method will determine if the computed head and power fall inside an avoidance zone. If so, the method will issue an error but not stop the run. If the method is called and there is no valid Unit Power, then the method is exited without performing any computations.

17.1.42 Frequency Regulation

This category depends on selecting the Unit Power Table method, although in the future it might be enabled for other power methods. The frequency regulation methods model the provision of the frequency regulation ancillary service, that is, how the reservoir can be made available to flexibly follow a load demand within a specified range during a certain period in order to affect the frequency of the generated power.

17.1.42.1 No Method (default)

This is the default, do-nothing method; no regulation is modeled.

17.1.42.2 Unit Frequency Regulation

NOTE: ALTHOUGH THE METHOD CAN BE SELECTED AND SLOTS ARE ADDED, THIS METHOD IS NOT YET IMPLEMENTED.

When frequency regulation is scheduled, it allows the unit to follow the real time load. Exactly what will happen in real time is unknowable. This results in two sets of values at scheduling time, nominally scheduled power and turbine release. It is uncertain if the real time operators will actually use the service. At present, we distinguish between the nominal “scheduled” power (and turbine release) that the regulation is allowed to depart from and the “expected” power generation (and turbine release) that will take place when regulation is allowed. Both are important. The scheduled power sets the baseline for regulation and should be communicated to the power dispatchers. The expected power and release are more useful for coordinating a plant with the rest of the system.

SLOTS ADDED BY THIS METHOD

Note, many of these slots have column or row dimensions based on the number of units. The rows/columns of these slots are expanded at the beginning of the run to match the value in the Number of Units slot. When first configuring this method, the user must enter the Number of Units, then run the model (stepping through 1 timestep is enough) to grow the slots to the right dimensions.

The following slots are instantiated when this method is selected:

☛ **UNIT REGULATION TABLE**

Type: TableSlot
Units: LENGTH, FLOW, POWER, FLOW, POWER
Description: This table (not visible to the user) represents the available regulation (both up and down in terms of flow and power) for each unit at each point in the Unit Power Table.
Information: This table is calculated using data in the Unit Power Table and the Avoidance Zone Table (if applicable) and could be calculated automatically at beginning of run in simulation and/or optimization. This table consists of a block of six

columns for each unit. The head and flow values should be the same as the Unit Power Table. The other four columns in the block are respectively Regulation Flow Up, Regulation Power Up, Regulation Flow Down, and Regulation Power Down. These values represent the minimum and maximum power achievable from the initial flow value without passing through an avoidance zone.We require that the heads in this table appear in the Unit Power Table as well.

I/O: Automatically calculated at beginning of run
Links: NA

☛ UNIT TWO SIDED REGULATION
Type: AggSeriesSlot
Units: POWER
Description: The value is the two sided frequency regulation for the unit at that timestep.
Information: There is one column for each unit.
I/O: Input or Output
Links: NA

☛ UNIT REGULATION UP
Type: AggSeriesSlot
Units: POWER
Description: The value is the frequency regulation up for the unit at that timestep.
Information: There is one column for each unit.
I/O: Input or Output
Links: NA

☛ UNIT REGULATION DOWN
Type: AggSeriesSlot
Units: POWER
Description: The value is the frequency regulation down for the unit at that timestep.
Information: There is one column for each unit.
I/O: Input or Output
Links: NA

☛ UNIT POSSIBLE REGULATION UP
Type: AggSeriesSlot
Units: POWER
Description: The value is the possible regulation up for the unit at that timestep.
Information: There is one column for each unit.
I/O: Input or Output
Links: NA

☛ Unit Possible Regulation Down

Type: AggSeriesSlot
Units: Power
Description: The value is the possible regulation down for the unit at that timestep.
Information: There is one column for each unit.
I/O: Input or Output
Links: NA

☛ Unit Flow Addition For Regulation

Type: AggSeriesSlot
Units: Flow
Description: The value is the additional release required to reach the frequency high point for the unit at that timestep.
Information: There is one column for each unit. This value is typically returned from optimization and set via a rule.
I/O: Rule
Links: NA

☛ Unit Flow Reduction For Regulation

Type: AggSeriesSlot
Units: Flow
Description: The value is the reduction in release required to reach the frequency low point for the unit at that timestep.
Information: There is one column for each unit. This value is typically returned from optimization and set via a rule.
I/O: Rule
Links: NA

☛ Unit Scheduled Mechanical Power

Type: AggSeriesSlot
Units: Power
Description: The value is the scheduled mechanical power generation, before subtracting regulation (or reactive power) for the unit at that timestep.
Information: There is one column for each unit.
I/O: Output only
Links: NA

☛ Unit Scheduled Turbine Release

Type: AggSeriesSlot
Units: Flow
Description: The value is the turbine flow which corresponds to the Unit Scheduled mechanical Power for the unit at that timestep.
Information: There is one column for each unit.

I/O: Output only
Links: NA

☛ Unit Operating Cost Per Regulation Table

Type: TableSlot
Units: VALUE ($)
Description: For each generating unit, this is the cost per unit of regulation.
Information: There is one row for each unit.
I/O: Input
Links: NA

☛ Unit Operating Cost

Type: AggSeriesSlot
Units: VALUE ($)
Description: This is the total cost of using a unit for regulation incurred during the run.
Information: There is one column for each unit.
I/O: Output
Links: NA

☛ Operating Cost

Type: AggSeriesSlot
Units: VALUE ($)
Description: The value is the sum of the unit operating costs.
Information: This is an existing slot with only one column.
I/O: Output only
Links: NA

☛ Regulation

Type: SeriesSlot
Units: POWER
Description: Total regulation for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Two Sided Regulation.
I/O: Output only
Links: Optional

☛ Plant Regulation Up

Type: SeriesSlot
Units: POWER
Description: Total regulation up for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Regulation Up.
I/O: Input or Output
Links: Linkable, typically to the Thermal object System Regulation Up

☛ Plant Regulation Down

Type: SeriesSlot
Units: Power
Description: Total regulation down for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Regulation Down.
I/O: Input or Output
Links: Linkable, typically to the Thermal object System Regulation Down

☛ Plant Possible Regulation Up

Type: SeriesSlot
Units: Power
Description: Total possible regulation up for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Possible Regulation Up.
I/O: Output only
Links: No

☛ Plant Possible Regulation Down

Type: SeriesSlot
Units: Power
Description: Total possible regulation down for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Possible Regulation Down.
I/O: Output only
Links: Optional

☛ Plant Flow Addition For Regulation

Type: SeriesSlot
Units: Flow
Description: Total additional turbine release required in order to reach the frequency regulation high point for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Flow Addition For Regulation.
I/O: Output only
Links: Optional

☛ Plant Flow Reduction For Regulation

Type: SeriesSlot
Units: Flow
Description: Total reduction in turbine release required in order to reach the frequency regulation low point for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Flow Reduction For Regulation.
I/O: Output only
Links: No

☛ Plant Scheduled Mechanical Power

Type: SeriesSlot
Units: Power
Description: Total scheduled mechanical power for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Scheduled Mechanical Power.
I/O: Output only
Links: No

☛ Plant Scheduled Turbine Release

Type: SeriesSlot
Units: Flow
Description: Total scheduled turbine release for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Scheduled Turbine Release.
I/O: Output only
Links: No

In Simulation, the Unit Power Table method will execute this method when the Unit Turbine Release is known. At this time, this method cannot be called if Unit Energy is specified (input or rules):

- Unit Scheduled Turbine Release[t,u] = Unit Turbine Release[t,u] - (Unit Flow Addition for Regulation[t,u] + Unit Flow Reduction For Regulation [t,u])/2
- Unit Scheduled Mechanical Power[t,u] = Unit Power Table(head[t], Unit Scheduled Turbine Release[t,u])
- Unit Regulation Up[t.u] = Unit Power Table(head[t], Unit Scheduled Turbine Release[t,u] + Unit Flow Addition for Regulation[t,u] /2)
- Unit Regulation Down [t,u] = Unit Power Table(head[t], Unit Scheduled Turbine Release[t,u]-Unit Flow Reduction For Regulation [t,u]/2)
- Unit Regulation [t,u] = max(Unit Regulation Up[t.u] , Unit Regulation Down [t,u])
- Unit Power[t,u] = Unit Scheduled Mechanical Power[t,u] + Unit Regulation Up[t.u]/2 - Unit Regulation Down [t,u]/2
- Calculate Unit Operating Cost[t,u] = timestep * Unit Operating Cost Per Regulation[u] * (Unit Regulation Up[t,u] + Unit Regulation Down[t,u])
- Compute plant level values as a sum of unit values

17.1.43 Disable Reservoir Processes

This category holds the Pass Inflows method which allows you to disable reservoir processes and pass inflows as though the reservoir wasn't there.

17.1.43.1 None

This is the default, no-action method. When this method is selected all physical processes are enabled and executed as appropriate.

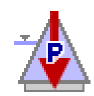
17.1.43.2 Pass Inflows

This method disables many of the reservoir physical processes and passes the inflows through the reservoir. This method can be used to temporarily remove the presence of a reservoir and compute flows as though they were unregulated, unconstrained or 'pre-project'.

No Slots are associated with this method.

This method does the following:

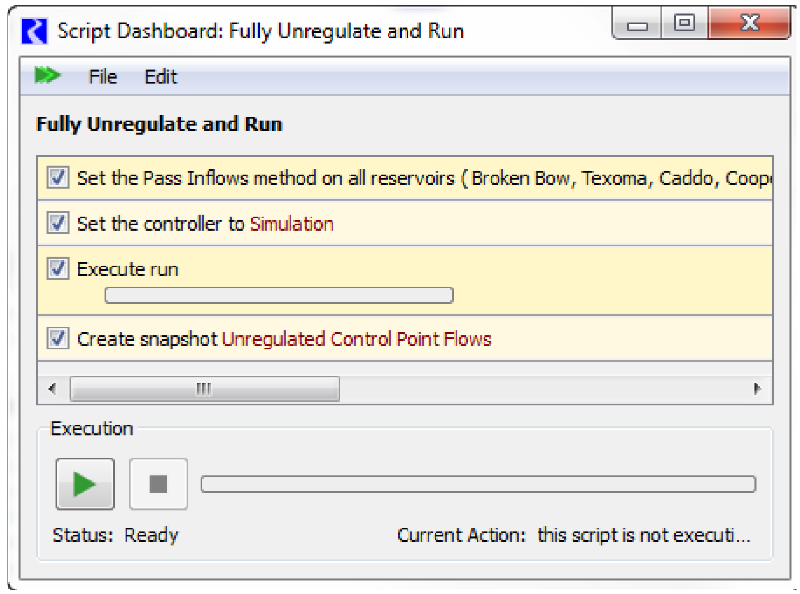
- **Changes the Workspace Icon**: When this method is selected, the icon on the workspace is modified to indicate the Reservoir is passing inflows.



- **Enables alternative dispatch method**: When this method is selected, ONLY the **Outflow Equals Sum of Inflows** dispatch method is available. This dispatch method is described HERE (Section 17.2.12).

Computation of flows for a Fully Unregulated system. To compute the flows at any point in the system without the effect of any reservoir, run two simulations. The first run computes the unregulated flows, the second computes the regulated flows, often using rulebased simulation. Results can be analyzed using snapshots. Simple scripts can be used to modify the system and make the runs. Following is the conceptual approach:

1. Compute unregulated flows by executing a script. Scripts are described **HERE (ScriptManagement.pdf, Section 1)**. A screenshot of a sample **Script Dashboard** is shown. It does the following:



- Set the Pass Inflows method on each reservoir.
- Set the run controller to **Simulation**
- Run the model. With the new method, the reservoirs pass inflows (including Hydrologic Inflow and Return Flow) but no other physical processes are made. When the reservoirs dispatch, they will set Outflow = ΣInflows, thus propagating the flows downstream.
- Create snapshots of desired slots.

2. You could then look at the results if desired.

3. Compute regulated flows by executing a script which does the following:

- Set the method on each reservoir to its normal regulated mode (**None** method in the **Disable Reservoir Processes** category).
- Set the run control to **Rulebased Simulation**
- Load the RBS ruleset set if necessary
- Run the model

4. At this point, the regulated results are in the objects and slots. They can be compared with the snapshots from the unregulated run.

The process above assumes two scripts, steps 1 and 3; these could be combined into one script that fully automates the runs if desired. With this approach it is easy for the user to run just the regulated system, the unregulated system or both.

This approach presented assumes that all reservoirs are unregulated. If only a portion of the system is unregulated, then you will need to decide how the other reservoirs behave, especially if they are dependent on the disabled reservoir. Additional script actions can be used to disable rules or set values in the system. The USACE approach to unregulated conditions is described **HERE (USACE_SWD.pdf, Section 4)**.

17.2 Dispatch Methods

The dispatch method used to solve the Level Power Reservoir, is dependent upon the selection made in the Hydrologic Inflow category and Disable Reservoir Processes categories.

17.2.1 solveMB_givenInflowHW

Solves the mass balance equation when Inflow and headwater (Pool Elevation) are known.

Required Knowns

- **Diversion**
- **Return Flow**
- **Inflow**
- **Pool Elevation**
- **Hydrologic Inflow**

(unless None is selected in the Hydrologic Inflow category)

Required Unknowns

- **Storage**
- **Outflow**
- **Energy**

First the dispatch method finds the Storage associated with the known Pool Elevation using the Elevation Volume Table. Then, if Pool Elevation is flagged as a TARGET, the method performs the appropriate target calculations. Next, the method checks for a linked Canal Object. If a Canal object is linked and has not yet solved, the dispatch method exits and waits for the Canal to solve before continuing.

If the above checks pass, the mass balance equation is solved as follows:

$$Outflow = Storage(t-1) - Storage + Inflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods. Inflow in the mass balance equation is the net inflow to the reservoir, including hydrologic inflows, return flows, and diversions if appropriate.

The spill calculation method is then executed followed by the power calculation method. If there are excess outflows from the power calculation method, the spill method is executed again to distribute the excess.

Energy in storage, spilled energy and future value calculations are performed if the user has selected them.

17.2.2 solveMB_givenOutflowHW

Solves the mass balance equation when Outflow and headwater (Pool Elevation) are known.

REQUIRED KNOWNS

☛ **DIVERSION**

☛ **RETURN FLOW**

☛ **OUTFLOW**

☛ **POOL ELEVATION**

☛ **HYDROLOGIC INFLOW**

(unless None is selected)

REQUIRED UNKNOWNS

☛ **STORAGE**

☛ **INFLOW**

First the dispatch method calculates the Storage using the Elevation Volume Table and the known Pool Elevation. Then it performs the appropriate target calculations if Pool Elevation is flagged as a TARGET (Note: if Outflow is flagged MAX CAPACITY a target operation cannot be performed). Next, the method checks for a linked Canal Object. If a Canal Object is linked and has not yet solved, the dispatch method waits for the Canal to solve before continuing. If Outflow is flagged MAX CAPACITY, it is set as the sum of the maximum spill and maximum turbine release.

The mass balance is carried out by the following equation:

$$Inflow = Storage - Storage(t-1) + Outflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods. Inflow calculated by the mass balance equation includes the effects of hydrologic inflow, diversion and return flow. The Inflow slot is adjusted accordingly to account for these additional terms.

The spill method is then executed. Next, the Energy is checked. If Energy is input, the power release method is executed instead of the power calculation method. Regardless of whether or not Energy is input, the Tailwater Base Value is checked. If the Tailwater Base Value (from a downstream reservoir) is not yet known, the method is exited and waits until the Tailwater Base Value is known. Next, the power calculation method is executed. Then the Turbine Release is calculated. After Turbine Release has been calculated, the spill method may be executed again if there are excess outflows. If the calculated Inflow is less than zero, and the Hydrologic Inflow And Loss method is selected, Inflow is set to 0.0 and Hydrologic Inflow Adjust is calculated.

Energy in storage, spilled energy and future value calculations are performed if the user has selected them. The dispatch method is then completed.

17.2.3 solveMB_givenInflowStorage

Solves the mass balance equation when Inflow and Storage are given.

Required Knowns

- ☛ **Diversion**
- ☛ **Return Flow**
- ☛ **Inflow**
- ☛ **Storage**
- ☛ **Hydrologic Inflow**

  (unless None is selected)

Required Unknowns

☛ **Pool Elevation**

☛ **Outflow**

☛ **Energy**

First, the method solves for Pool Elevation using the known Storage and the Elevation Volume Table. Then, if Storage is flagged as a TARGET, the appropriate target calculations are performed. If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. Next, the method checks for a linked Canal Object. If a Canal Object is linked and has not yet solved, the dispatch method waits for the Canal to solve before continuing.

The mass balance is carried out by the following equation:

$$Outflow = Storage(t-1) - Storage + Inflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods. Inflow in the mass balance equation is the net inflow to the reservoir, including hydrologic inflows, return flows, and diversions if appropriate.

The spill calculation method is then executed followed by the power calculation method. If there are excess outflows from the power calculation method, the spill method is executed again to distribute the excess.

Energy in storage, spilled energy and future value calculations are performed if the user has selected them. The dispatch method is then completed.

17.2.4 solveMB_givenOutflowStorage

Solves the mass balance equation when Outflow and Storage are known.

Required Knowns

- **Diversion**
- **Return Flow**
- **Outflow**
- **Storage**
- **Hydrologic Inflow**

  (unless None has been selected)

Required Unknowns

- **Pool Elevation**
- **Inflow**

First the dispatch method calculates the Pool Elevation using the Elevation Volume Table and the known Storage. Then the appropriate target calculations are performed if Storage is flagged as a TARGET (if Outflow is flagged MAX CAPACITY a target operation cannot be performed on Storage). If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. Next the method checks for a linked Canal Object. If a Canal Object is linked and has not yet solved, the dispatch method waits for it to solve before continuing. If Outflow is flagged MAX CAPACITY, it is set as the sum of the maximum spill and maximum turbine release.

The mass balance is carried out by the following equation:

$$Inflow = Storage - Storage(t-1) + Outflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods. The Inflow calculated by the mass balance equation includes the effects of hydrologic inflow, diversion and return flow. The Inflow slot is adjusted accordingly to account for these additional terms.

The spill method is then executed followed by the power calculation method. If Energy is input, the power release method is executed instead of the power calculation method. After Turbine Release has been calculated, the spill method may be executed again if there are excess outflows.

Energy in storage, spilled energy and future value calculations are performed if the user has selected them. The dispatch method is then completed.

17.2.5 solveMB_givenInflowOutflow

Solves the mass balance equation when Inflow and Outflow are known.

Required Knowns

- **Diversion**
- **Return Flow**
- **Inflow**
- **Outflow**
- **Hydrologic Inflow**

(unless None is selected)

Required Unknowns

- **Pool Elevation**
- **Storage**

First, the previous timestep's Pool Elevation and Storage are checked. If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. If the Canal object is linked to the reservoir and has not yet solved, the dispatch method is exited and waits for the Canal to solve before continuing.

If Outflow is flagged MAX CAPACITY, it is solved for by the function getMaxOutGivenIn(). This function uses a convergence algorithm detailed HERE (Appendix A: Reservoir Convergence). The Outflow slot is set equal to the computed maximum outflow value.

If the Outflow slot is set by the Surcharge Release flag, the surcharge release is computed and the Outflow slot is set equal to surcharge release for all timesteps in the forecast period (the surcharge release methods compute a surcharge release forecast). Additional information on dispatching when using the Surcharge Release Flag can be found **HERE (USACE_SWD.pdf, Section 3.3.3)**.

The mass balance is then carried out as follows:

$$Storage = Storage(t-1) + Inflows - Outflows$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods. The inflow term includes the effects of hydrologic inflow, diversion and return flow. Since the evaporation, bank storage, seepage and precipitation terms depend upon the Storage value, the mass balance solution is an iterative process. (See note below.) Pool Elevation is calculated using the Elevation Volume Table and the calculated Storage value.

If the Outflow is less than zero, the method is exited.

The spill method is then executed followed by the power calculation method. If Energy is input, the power release method is executed instead of the power calculation method. After Turbine Release has been calculated, the spill method may be executed again if there are excess outflows.

Energy in storage, spilled energy and future value calculations are performed if the user has selected them. The dispatch method is then completed.

NOTE: In specific cases when Storage is very close to zero, the iterative mass balance solution will follow one of two possible processes. The first process allows the loop to iterate in the negative storage range before final convergence. Convergence in this situation is typically quite rapid, and in instances when the final storage is, in fact, negative, an error is posted stating that the outflow is too large to be physically possible. This process is invoked if an additional row is appended to the Elevation Volume Table specifying a negative storage value within which the loop can iterate. If specifying and allowing negative storage values in the iteration is not desired, no negative storage value should be appended to the Elevation Volume Table. In this second process, the algorithm uses storage equals zero whenever it is in the negative storage range. If the outflow is really too great to be physically possible, the algorithm will keep iterating until it reaches maximum iterations. If this happens, RiverWare does a final mass balance check at the storage equals zero point and posts an error stating that the outflow is too large.

17.2.6 solveMB_givenEnergyInflow

Solves the mass balance equation when Energy and Inflow are given.

Required Knowns

☛ **DIVERSION**

☛ **RETURN FLOW**

☛ **INFLOW**

☛ **ENERGY**

☛ **HYDROLOGIC INFLOW**

(unless None is selected)

REQUIRED UNKNOWNS

☛ **POOL ELEVATION**

☛ **STORAGE**

First, the previous timestep's Pool Elevation and Storage are checked. If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. If the Canal object is linked and has not yet solved, the dispatch method is exited and waits for the Canal to solve before continuing. Then the selected spill and tailwater methods are executed. The selected power release method is executed and Outflow is set as Turbine Release plus Spill.

The mass balance equation used to solve for Storage is:

$$Storage = Storage(t-1) + Inflow - Outflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods. The inflow term includes the effects of hydrologic inflow, diversion and return flow. Since the evaporation, bank storage, seepage and precipitation terms depend upon the Storage value, the mass balance solution is an iterative process. (See note below.) Pool Elevation is then calculated using the Elevation Volume Table and the calculated Storage value.

An iterative process is then executed in which the storage solution described above iterates within another iterative loop which solve for Outflow. When a new Storage value is calculated it is used to recompute Spill, Operating Head, and Turbine Release. Outflow is then set as the sum of Spill and Turbine Release. The iteration process continues until the solution stabilizes. When the iterations are complete, the power calculation method is executed.

Energy in storage, spilled energy and future value calculations are performed if the user has selected them. The dispatch method is then completed.

NOTE: In specific cases when Storage is very close to zero, the iterative solution for Storage will follow one of two possible processes. The first process allows the loop to iterate in the negative storage range before final convergence. Convergence in this situation is typically quite rapid, and in instances when the final storage is, in fact, negative, an error is posted stating that the outflow is too large to be physically possible. This process is invoked if an additional row is appended to the Elevation Volume Table specifying a negative storage value within which the loop can iterate. If specifying and allowing negative storage values in the iteration is not desired, no negative storage value should be appended to the Elevation Volume Table. In this second process, the algorithm uses storage equals zero whenever it is in the negative storage range. If the outflow is really too great to be physically possible, the algorithm will keep iterating until it reaches maximum iterations. If this happens, RiverWare does a final mass balance check at the storage equals zero point and posts an error stating that the outflow is too large.

17.2.7 solveMB_givenEnergyStorage

Solves the mass balance equation when Energy and Storage are known.

Required Knowns

- **Diversion**
- **Return Flow**
- **Energy**
- **Storage**
- **Hydrologic Inflow**

 (unless None is selected)

Required Unknowns

- **Pool Elevation**
- **Inflow**

First, the method calculates Pool Elevation using the Elevation Volume Table and the known Storage value. If Storage is flagged as a TARGET, the reservoir is overdetermined and a

RiverWare™ error is posted and the run is aborted. If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. If the Canal object is linked and has not yet solved, the dispatch method is exited and waits for the Canal to solve before continuing. Spill is then calculated according to the user specified method. The evaporation, bankstorage, and seepage calculations are then performed.

An iterative process is then used to calculate Outflow. The Tailwater method is executed to solve for operating head. The value of operating head is used to determine the Turbine Release from the user selected power release method. Outflow is then set as the sum of Turbine Release and Spill. This process is then repeated until the value of Outflow converges.

Inflow is then calculated from the mass balance equation:

$$Inflow = Storage - Storage(t-1) + Outflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods. The inflow term calculated by the mass balance equation includes the effects of hydrologic inflow, diversion and return flow. The Inflow slot is adjusted accordingly to account for these additional terms.

Power, Energy in storage, spilled energy and future value calculations are performed if the user has selected them.

17.2.8 solveMB_givenEnergyHW

Solves the mass balance equation when Energy and headwater (Pool Elevation) are known.

Required Knowns

- ☛ **Diversion**
- ☛ **Return Flow**
- ☛ **Pool Elevation**
- ☛ **Energy**
- ☛ **Hydrologic Inflow**

(unless None is selected)

REQUIRED UNKNOWNS

☛ STORAGE

☛ INFLOW

First the Storage is computed using the Elevation Volume Table and the known Pool Elevation. If Pool Elevation is flagged as a TARGET, the reservoir is overdetermined and a **RiverWare**™ error is posted and the run is aborted. If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. If the Canal object is linked and has not yet solved, the dispatch method is exited and waits for the Canal to solve before continuing. Spill, evaporation, bankstorage and seepage are then calculated according to the user specified methods.

An iterative process is used to calculate Outflow. The Tailwater method is executed to solve for operating head. The value of operating head is used to determine the Turbine Release from the user selected poser release method. Outflow is then set as the sum of Turbine Release and Spill. This process is then repeated until the value of Outflow converges.

Inflow is then calculated from the mass balance equation:

$$Inflow = Storage - Storage(t-1) + Outflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods. The inflow term calculated by the mass balance equation includes the effects of hydrologic inflow, diversion and return flow. The Inflow slot is adjusted accordingly to account for these additional terms.

Power, Energy in storage, spilled energy and future value calculations are performed if the user has selected them. The dispatch method is then completed.

17.2.9 solveMB_givenInflowRelease

This dispatch method executes when the following dispatch conditions are met.

REQUIRED KNOWNS

- **DIVERSION**
- **RETURN FLOW**
- **INFLOW**
- **TURBINE RELEASE**
- **HYDROLOGIC INFLOW**

  (unless None is selected)

REQUIRED UNKNOWNS

- **ENERGY**
- **POOL ELEVATION**
- **STORAGE**
- **OUTFLOW**

First, the previous timestep's Pool Elevation and Storage are checked. If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. The dispatch method then checks if a Canal Object is linked. If it is linked, a **RiverWare**™ error is flagged and the run is aborted. The Canal object cannot be solved with this dispatch method. A check is also performed to see if Spill is input. If Spill is input, a **RiverWare**™ error is posted and the run is aborted. Only Regulated Spill and Bypass may be input for this method.

If the Turbine Release is flagged Unit Values (U), the Unit Turbine Release subslots are summed to calculate and set the Turbine Release slot. If all the Unit Turbine Release slots are NaN, an error is issued. If the Unit Turbine Release slot is not visible because the Unit Power Table method is not selected, an error is issued.

If there is an unregulated spillway crest (because an unregulated spill method is selected) the method computes the upper limit for unregulated spill to prevent spilling too much water to drop the reservoir below the crest. Typically, this limit is computed as the volume of water above the crest converted to a flow. See the Unregulated Spill method for details.

Then mass balance occurs. The Outflow is computed as the sum of turbine release and spill inputs (if any) according to the user selected methods and the equation:

$$Outflow = Spill + Release$$

Storage is solved for as follows:

$$Storage = Storage(t-1) + Inflow - Outflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods. The inflow term includes the effects of hydrologic inflow, diversion and return flow. Since the evaporation, bank storage, seepage and precipitation terms depend upon the Storage value, the mass balance solution is an iterative process. (See note below.) Pool Elevation is calculated using the Elevation Volume Table and the calculated Storage value.

After the Storage is computed, mass balance process is compete, the spill calculation method is executed. Unregulated Spill takes up any error in the solution process. For this reason, it may not conform to the Unregulated Spill Table.

Power, Energy in Storage, spilled energy and future value calculations are performed if the user has selected them. The dispatch method is then completed.

NOTE: In special cases when Storage is very close to zero, the iterative mass balance solution will follow one of two possible processes. The first process allows the loop to iterate in the negative storage range before final convergence. Convergence in this situation is typically quite rapid, and in instances when the final storage is, in fact, negative, an error is posted stating that the outflow is too large to be physically possible. This process is invoked if an additional row is appended to the Elevation Volume Table specifying a negative storage value within which the loop can iterate. If specifying and allowing negative storage values in the iteration is not desired, no negative storage value should be appended to the Elevation Volume Table. In this second process, the algorithm uses storage equals zero whenever it is in the negative storage range. If the outflow is really too great to be physically possible, the algorithm will keep iterating until it reaches maximum iterations. If this happens, RiverWare does a final mass balance check at the storage equals zero point and posts an error stating that the outflow is too large.

17.2.10 solveMB_givenInflowOutflowStorage

Solves for Hydrologic Inflow when the Solve Hydrologic Inflow method is selected. Inflow, Outflow, and Storage must be known for this method to solve.

Required Knowns

☛ Diversion

☛ Return Flow

☛ Outflow

☛ Storage

☛ Inflow

Required Unknowns

☛ Pool Elevation

☛ Hydrologic Inflow

First the method checks if Outflow is flagged MAX CAPACITY. The MAX CAPACITY flag cannot be used when both Inflow and Storage are known. Pool Elevation is then calculated using the Elevation Volume Table and the known Storage value. If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. If the Storage slot is flagged TARGET a **RiverWare**™ error is posted and the run is aborted. If the Canal object is linked and has not yet solved, the dispatch method is exited and waits for the Canal to solve before continuing. The user selected evaporation, bankstorage, and seepage methods are then executed.

The the hydrologic inflow is found with the following equation (all values are converted to flows):

$$HydrologicInflow = Storage - Storage(-1) + Outflow - Inflow$$

This equation may or may not contain other elements reflecting the user method choices (these include Evaporation (+), Precipitation(-), Seepage(+), Bankstorage(+), Return Flow(-), Diversion (+), and Flow TO/FROM Pumped Storage). Spill and release are then found according to the user selected methods.

If Outflow is determined to be less than zero, the method is exited.

The spill method is then executed followed by the power calculation method. If Energy is input, the power release method is executed instead of the power calculation method. After Turbine Release has been calculated, the spill method may be executed again if there are excess outflows.

Energy in storage, spilled energy and future value calculations are performed if the user has selected them.

17.2.11 solveMB_givenInflowOutflowHW

Solves for Hydrologic Inflow when the Solve Hydrologic Inflow method is selected. Inflow, Outflow, and headwater (Pool Elevation) must be known for this method to solve.

Required Knowns

- **Diversion**
- **Return Flow**
- **Outflow**
- **Pool Elevation**
- **Inflow**

Required Unknowns

- **Storage**
- **Hydrologic Inflow**

First the method checks if Outflow is flagged MAX CAPACITY. The MAX CAPACITY flag cannot be used when both Inflow and Storage are known. Storage is then calculated using the Elevation Volume Table and the known Pool Elevation value. If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. If the Pool Elevation slot is flagged TARGET, a **RiverWare**™ error is posted and the run is aborted. If a Canal Object is linked and has not yet solved, the dispatch method is exited and waits for the Canal to solve before continuing. The user selected evaporation, bankstorage, and seepage methods are then executed.

The the hydrologic inflow is found with the following equation (all values are converted to flows):

$$HydrologicInflow = Storage - Storage(-1) + Outflow - Inflow$$

This equation may or may not contain other elements reflecting the user method choices (these include Evaporation (+), Precipitation(-), Seepage(+), Bankstorage(+), Return Flow(-), Diversion (+), and Flow TO/FROM Pumped Storage).

If Outflow is determined to be less than zero, the method is exited.

The spill method is then executed followed by the power calculation method. If Energy is input, the power release method is executed instead of the power calculation method. After Turbine Release has been calculated, the spill method may be executed again if there are excess outflows.

Energy in storage, spilled energy and future value calculations are performed if the user has selected them. The dispatch method is then completed.

17.2.12 Outflow Equals Sum of Inflows

This dispatch method is only available when the **Pass Inflows** method HERE (Section 17.1.43.2) is selected. It has the following dispatch conditions:

Required Knowns

☛ **Inflow**

Required Unknowns

☛ **Outflow**

This method does the following:

An error is issued if any of the following are linked, specified as inputs, and/or are not zero:

- Canal Flow,
- Diversion,
- Flow TO Pumped Storage,
- Flow FROM Pumped Storage
- Seepage

An error is issued if:

- Any flags are set on Outflow (an over-determination error will be issued).
- Water Quality is enabled on this reservoir

Inflow Sum and Total Inflows are computed as usual as described HERE (Total Inflows)

Finally:

Outflow = Inflow + Return Flow + Hydrologic Inflow + Hydrologic Inflow Adjust + Hydrologic Inflow Forecast

Then the dispatch method finishes successfully and exits.

Note: Hydrologic Inflow disaggregation described HERE (Section 17.1.17) and forecasting described HERE (Section 17.1.16) is performed at beginning of the run or the timestep. Therefore, these inflows to the reservoir are included in the outflow.

But, none of the other physical process methods are executed even though they may have method selections and slots visible. These include:

- Precipitation and Evaporation
- Spill
- Energy in Storage
- Seepage
- Bank Storage
- Tailwater
- Operating Level
- Power

18. Pipe Junction

This object models a Pipe Junction. It simulates the split or junction of pressurized flow.

General Slots

☞ Flow 1

Type: SeriesSlot
Units: Flow
Description: Flow into or out of the object
I/O: Input, Output, or set through a link
Links: Usually linked

☞ Flow 2

Type: SeriesSlot
Units: Flow
Description: Flow into or out of object
I/O: Input, Output, or set through a link
Links: Usually linked

☞ Flow 3

Type: SeriesSlot
Units: Flow
Description: Flow into or out of the object
I/O: Input, Output, or set through a link
Links: Usually linked

The mass balance equation is:

$$Flow3 = Flow1 + Flow2$$

The Pipe Junction can be used to split flows similar to a bifurcation or bring flows together similar to a confluence. The linking necessary for these two setups is slightly different because of the signs of the flows. To join flows, the two incoming flows should be linked to

Flow 1 and Flow 2, the outgoing flow should be linked to Flow 3. Optionally, select the “Solve Flow 3” method to force the object to solve only in one direction, if necessary.

To model a spliting of flows, the incoming flow should be linked to Flow 3 and the two outgoing flows should be linked to Flow 1 and Flow 2. Optionally, select the “Solve Flow 1 or Flow 2 Only”, “Solve Flow 1 Only”, or “Solve Flow 2 Only” method to force the object to solve only in one direction, if necessary.

18.1 User Methods

18.1.1 Pipe Junction Solution Direction

The methods in this category are used to control the available dispatch methods and therefore which direction the pipe junction is solving. The none default methods may be selected in Rulebased Simulation when you need to control this direction.

18.1.1.1 Solve Flow 1, Flow 2, or Flow 3

This is the default method and should remain selected for basic simulation. There are no slots or calculations specifically associated with this method. It simply allows all dispatch methods to be active.

18.1.1.2 Solve Flow 1 or Flow 2 Only

This method may need to be selected in Rulebased Simulations if the Pipe Junction is solving in one direction only. When this method is selected, only the “Solve Flow 1” or “Solve Flow 2” dispatch methods are available.

18.1.1.3 Solve Flow 1 Only

This method may need to be selected in Rulebased Simulations if you want the Pipe Junction to solve in one direction only. When this method is selected, only the “Solve Flow 1” dispatch method is available.

18.1.1.4 Solve Flow 2 Only

This method may need to be selected in Rulebased Simulations if you want the Pipe Junction to solve in one direction only. When this method is selected, only the “Solve Flow 2” dispatch method is available.

18.1.1.5 Solve Flow 3 Only

This method may need to be selected in Rulebased Simulations if you want the Pipe Junction to solve in one direction only. When this method is selected, only the “Solve Flow 3” dispatch method is available.

18.1.2 Pipe Junction Hydraulics

The user selectable methods for **Pipe Junction Hydraulics** include “None” and “Propagate Head”. “None” is the default method and performs no calculations. “Propagate Head” will use the input/linked value for the known head to calculate the unknown heads.

18.1.2.1 None

This is the default method and performs no calculations.

18.1.2.2 Propagate Head

This method will use the values for the known head to calculate the unknown heads.

Slots Specific To This Method

☞ Flow 1 Head

Type: SeriesSlot
Units: Length
Description: Head at the Pipe Junction Flow 1.
I/O: Input or linked
Links: Usually linked to the head slot on another object

☞ Flow 2 Head

Type: SeriesSlot
Units: Length
Description: Head at the Pipe Junction Flow 2.
I/O: Output
Links: Usually linked to the head slot on another object

☞ Flow 3 Head

Type: SeriesSlot
Units: Length
Description: Head at the Pipe Junction Flow 3.
I/O: Output
Links: Usually linked to the head slot on another object

☞ **CONVERGENCE**

Type: ScalarSlot

Units: NO UNITS

Description: A value ranging from 0 to 1.0 used for convergence in all iterative calculations

Information: Input only, defaults to 0.001 if not input

I/O: Not linkable

The head must be specified for one of the known flow values. For example, if Flow 3 and Flow 2 are known, a value must be known (input, rules, or linked) for either flow 3 Head or Flow 2 Head. If both are specified, they must be within convergence of each other or an error message will be posted. If neither are known, the method will stop and wait for a value to be known. The known head slot will be used to solve for the other two unknown head slots. All three head slots will have the same head value.

18.2 Dispatch Methods

This object solves for Flow 1, Flow 2, or Flow 3 depending on the knowns and unknowns.

18.2.1 Solve Flow 1

The required knowns and unknowns (at the current timestep) are listed below.

REQUIRED KNOWNS

- **FLOW 2**
- **FLOW 3**

REQUIRED UNKNOWNS

- **FLOW 1**

If a Hydraulic Calculation method is selected, the dispatch method will execute the selected method and set the appropriate Head slots. Next, the flows are set according to the equation:

$$Flow1 = Flow3 - Flow2$$

18.2.2 Solve Flow 2

The required knowns and unknowns (at the current timestep) are listed below.

REQUIRED KNOWNS

- **FLOW 1**
- **FLOW 3**

REQUIRED UNKNOWNS

- **FLOW 2**

If a Hydraulic Calculation method is selected, the dispatch method will execute the selected method and set the appropriate Head slots. Next, the flows are set according to the equation:

$$Flow2 = Flow3 - Flow1$$

18.2.3 Solve Flow 3

The required knowns and unknowns (at the current timestep) are listed below.

Required Knowns

☞ Flow 1

☞ Flow 2

Required Unknowns

☞ Flow 3

If a Hydraulic Calculation method is selected, the dispatch method will execute the selected method and set the appropriate Head slots. Next, the flows are set according to the equation:

$$Flow3 = Flow1 + Flow2$$

19. Pipeline

This object models fully-pressurized flow through a pipeline.

General Slots

(slots which always appear for this object)

☞ INFLOW

Type: SeriesSlot

Units: FLOW

Description: Flow rate at the entrance of the object.

I/O: Can be input, output, or set by propagation across a link

Links: Usually linked to the outflow slot on another object.

☞ OUTFLOW

Type: Series Slot

Units: FLOW

Description: Flow rate at the exit of the object.

I/O: Can be input, output, or set by propagation across a link

Links: Usually linked to the inflow on another object.

19.1 User Methods

19.1.1 Pipeline Solution Direction

This method category is used to specify the direction the Pipeline solves (downstream given inflow or upstream given outflow). For basic Simulation runs this does not make a difference and the default method should remain selected. However, in Rulebased Simulation the user may need to limit the Pipeline to downstream solution only.

19.1.1.1 Solve Upstream or Downstream

This is the default method and should remain selected for basic Simulation. There are no slots or calculations specifically associated with this method. It simply allows all dispatch methods to be active.

19.1.1.2 Solve Downstream Only

This method may need to be selected in Rulebased Simulations where the Pipeline can solve in the downstream direction only. When this method is selected, only the solveOutflowGivenInflow dispatch method is active.

19.1.2 Head Loss

The user selectable methods for **Head Loss** include “None” and “Hazen-Williams Equation”. “None” is the default method and makes no other calculations. “Hazen-Williams Equation” will calculate Inflow Head given Outflow Head or Outflow Head given Inflow Head. The selected method executes from the dispatch method.

19.1.2.1 None

This is the default method and makes no calculations.

19.1.2.2 Hazen-Williams Equation

This method calculates Inflow Head given Outflow Head (and other inputs) or Outflow Head given Inflow Head (and other inputs). These inputs include pipe diameter, pipe length, the C Value of the Pipeline, and minor losses within the Pipeline.

In this method, either Inflow Head or Outflow Head must be given (either set as an input, by rules, or set through a link) in order to calculate the other. If Inflow Head is input or set through a link, Outflow Head is calculated. If Outflow Head is input or set through a link, Inflow Head is calculated.

The following slots will be instantiated when using the “Hazen-Williams Equation” method:

Slots Specific To This Method

☞ Inflow Head

Type: SeriesSlot

Units: LENGTH

Description: Inflow Head of the Pipeline object. This is equal to the total head, which includes pressure, elevation, and velocity components. This value likely comes from a link or is calculated.

I/O: Input or Output

Links: Usually linked to the head slot on another object

☞ Outflow Head

Type: SeriesSlot

Units: LENGTH

Description: Outflow Head of the Pipeline object. This is equal to the total head, which includes pressure, elevation, and velocity components. This value likely comes from a link or is calculated.

I/O: Input or Output

Type: Usually linked to the head slot on another object

☞ DIAMETER

Type: ScalarSlot

Units: LENGTH

Description: Diameter of the Pipeline. It is assumed constant over its length. It must be greater than zero.

I/O: Input only

Links: not linkable

☞ VELOCITY

Type: SeriesSlot

Units: VELOCITY

Description: Velocity of the water within the Pipeline.

I/O: Output only

Links: not linkable

☞ LENGTH

Type: ScalarSlot

Units: LENGTH

Description: Length of the Pipeline. The length must be greater than or equal to zero.

I/O: Input only

Links: not linkable

☞ C VALUE

Type: ScalarSlot

Units: NOUNITS

Description: Hazen-Williams C Value of the Pipeline. The C Value depends on the pipe material, pipe age, tuberculation, lining, etc.... Values can be found in a hydraulics text and vary between 80-150.

I/O: Input only

Links: not linkable

☞ Minor Losses

Type: ScalarSlot

Units: LENGTH

Description: Minor Losses at the Pipeline. The value defaults to zero if less than zero or not input.

I/O: Input only

Links: not linkable

The head loss will be calculated using the Hazen-Williams formula:

$$V = k \cdot C \cdot (R_h)^{0.63} \cdot (S)^{0.54}$$

Rh is the hydraulic radius. For a circular pipe:

$$R_h = \frac{Diameter}{4} = \frac{D}{4}$$

V is the velocity of the water within the pipeline:

$$V = Velocity = \frac{(Inflow = Outflow = Q)}{Area} = \frac{Q}{\left(\frac{\pi D^2}{4}\right)}$$

C is the user input Hazen-Williams C Value of the pipe:

$$C = \text{C Value}$$

S is the slope of the energy grade line:

$$S = \frac{\text{Head Loss}}{Length}$$

Note that the empirically-derived Hazen-Williams formula is unit specific. Published in literature, k for British units is 1.318 and 0.85 for SI units. In RiverWare, the formula uses S.I. units, in which head loss has units of meters, Length has units of meters, Velocity has units of meters/second, the C Value has no units, and Diameter has units of meters. Unfortunately, k =1.318 for British and k = 0.85 for SI are not equivalent when converted. As a result, RiverWare will use the more common and more significant value of k = 1.318. In SI units, this is equivalent to 0.8492.

Rearranging the Hazen-Williams formula, head loss can be solved for:

$$HeadLoss = Length \cdot \left(\frac{\left(\frac{4Q}{\pi D^2}\right)}{kC \cdot \left(\frac{D}{4}\right)^{0.63}} \right)^{\left(\frac{1}{0.54}\right)}$$

Depending on inputs, Outflow Head or Inflow Head is set as:

$$OutflowHead = InflowHead - HeadLoss - MinorLosses$$

OR

$$InflowHead = OutflowHead + HeadLoss + MinorLosses$$

19.2 Dispatch Methods

This object solves for inflow or outflow, whenever the other is known. If the Solve Upstream or Downstream method is selected, then all dispatch methods are active. However, if Solve Downstream Only is selected, then only the solveOutflowGivenInflow method is active.

19.2.1 solveOutflowGivenInflow

Solves for outflow.

Required Knowns

☞ **Inflow**

Required Unknowns

☞ **Outflow**

If an inflow head is linked, the dispatch method will wait for a value in this slot before solving. The equation for this dispatch method is:

$$Outflow = Inflow$$

19.2.2 solveInflowGivenOutflow

Solves for Inflow.

Required Knowns

☞ **Outflow**

Required Unknowns

☞ **Inflow**

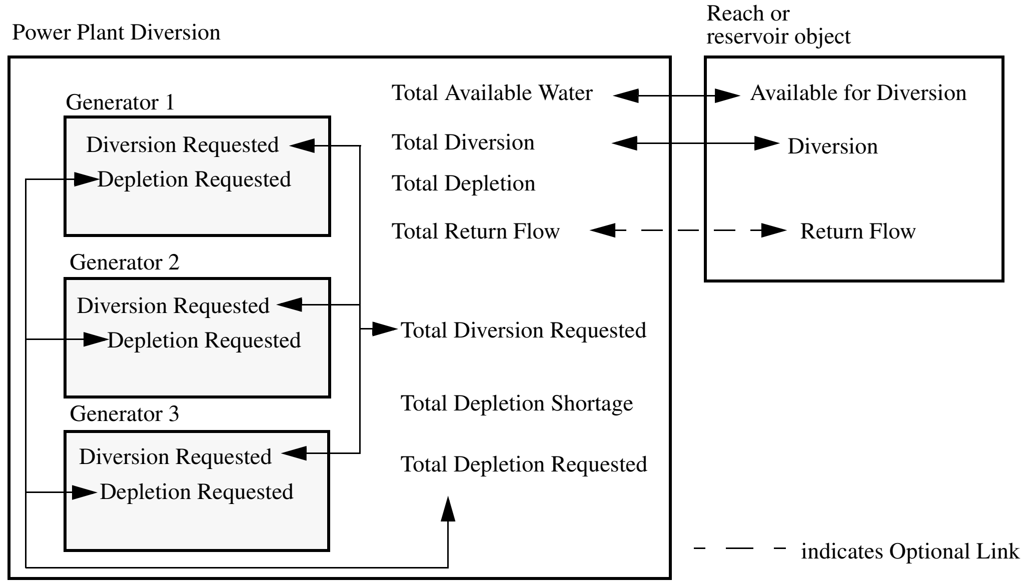
If an outflow head is linked, the dispatch method will wait for a value in this slot before solving. The equation for this dispatch method is:

$$Inflow = Outflow$$

20. Power Plant Diversion

The Power Plant is a collection of individual generators (described HERE (Section 13). A single diversion is used to serve all generators in the plant. Thus, the plant level computations divert water from a reach or reservoir to meet the Total Diversion Requested limited by the Total Available Water. The plant also computes the depletion and consumption of water and diversion and depletion shortage. If there is a shortage, then there is no need to allocate that shortage to individual generators.

Following is a common linking structure of the Power Plant Diversion, Generators and reach/reservoir.



The individual Generator requests are summed in the **Total Diversion Requested** slot on the Power Plant Diversion.

Note, it is not technically necessary to use Generator elements with this linking structure. If desired, **Total Diversion Requested** and **Total Depletion Requested** can be input directly by the user. However, it is often desirable to use one or more Generator objects for this purpose.

There are no user methods for this object.

General slots

(slots which always appear for this object)

☞ Total Available Water

Type: Series Slot

Units: FLOW

Description: signifies the water available for diversion from the reservoir or reach to which the diversion is connected

Information: Set by link propagation from the diverted object. The user is responsible for linking this slot with the object from which the water is diverted.

I/O: Optional; Could be input if not set via link.

Links: May be linked.

☞ Total Depletion

Type: Series Slot

Units: FLOW

Description: The total amount consumed.

Information: Computed typically as Total Diversion minus Total Return Flow

I/O: Output only

Links: Usually not linked

☞ Total Depletion Requested

Type: Multislot

Units: FLOW

Description: a multislot showing the Depletion Requested for each Generator as well as the sum of all requested depletions.

Information:

I/O: Optional; Can be input when there are no Generator elements and it is not linked.

Links: Usually not linked to other objects

☞ Total Diversion

Type: Series Slot

Units: FLOW

Description: denotes the total water diverted to the Water Users

Information:

I/O: Optional; must be specified if Total Diversion Requested is not given.

Links: Typically linked to the Diversion slot of a reach or a reservoir.

☞ Total Diversion Requested

Type: MultiSlot

Units: FLOW

Description: a multislot showing the Diversion Requested for each Generator as well as the sum of all requested diversions.

I/O: Optional; Can be input when there are no Generator elements and it is not linked.

Links: Linked to the individual generators. It can also be linked to other objects as necessary.

☞ Total Return Flow

Type: Series Slot

Units: FLOW

Description: represents the portion of the water diverted which is not consumed

Information: It is the user's responsibility to specify what is done with this data. If it is not linked to another object (i.e. back to the reach/reservoir Return Flow slot), it is lost from the system.

I/O: Output only

Links: May be linked, for example, to the Return Flow slot on any object or the Inflow slot on a Groundwater Storage object.

☞ Total Depletion Shortage

Type: Series Slot

Units: Flow

Description: Represents the difference between Total Depletion Requested and Total Depletion.

Information: Computed as Total Depletion Requested - Total Depletion

I/O: Output Only

Links: Not Linkable

☞ Total Diversion Shortage

Type: Series

Units: Flow

Description: The difference between Diversion Requested and Diversion

I/O: Output Only

Links: Not linkable

20.1 Dispatch Methods

The Power Plant Diversion solves given either diversion requests and available water (most common) or just diversion.

20.1.1 Solve given Diversion Request and Available

This dispatch method will solve when diversion requests and the available water are known. It computes the total diversion and depletion (consumption).

The **Total Diversion Requested** is compared with the **Total Available Water** slot. If the total request can be met, the **Total Diversion** slot is set to that amount; if not, the **Total Diversion** slot is set to the **Total Available Water** amount. If a total deficit occurs in the **Diversion**, the **Return Flows** are recalculated based on the ratio of the **Total Depletion Requested** to the **Total Diversion Requested**. The **Total Depletion Shortage** is computed as the difference between **Total Depletion Requested** and **Depletion**.

Required Knowns

- **Total Diversion Requested**
- **Total Available Water**

Required Unknowns

- **Total Diversion**
- **Total Depletion**

A general description of the interaction among the Power Plant Diversion, Generator(s), and Reach/Reservoir is given below.

1. Reach/Reservoir **Available For Diversion** is set.
2. This value propagates to Power Plant Diversion: **Total Available Water**.
3. **Solve given Diversion Request and Available** dispatches on the Power Plant Diversion object.
4. Within the dispatch method, **Total Diversion** and **Total Return Flow** are calculated.
5. The values propagate to the reach or reservoir and that object re-solves.

The **Solve given Diversion Request and Available** does the following

If **Total Depletion Requested** is not specified, it is set equal to the **Total Diversion Requested**.

If **Total Diversion Requested** is greater than or equal to the **Total Available Water**:

$$\textbf{Total Diversion} = \textbf{Total Available Water}$$

$$\text{Total Return Flow} = \text{Total Diversion} \times \left(1 - \frac{\text{Total Depletion Requested}}{\text{Total Diversion Requested}}\right)$$

If the **Total Diversion Requested** is less than **Total Available Water**:

$$\text{Total Diversion} = \text{Total Diversion Requested}$$

$$\text{Total Return Flow} = \text{Total Diversion - Total Depletion Requested}$$

Total Depletion is set as the difference between **Total Diversion** and **Total Return Flow**.

Total Depletion Shortage is computed as **Total Depletion Requested** minus **Total Depletion**.

Total Diversion Shortage is computed as **Total Diversion Requested** minus **Total Diversion**.

20.1.2 Solve given Diversion

This dispatch method will solve when diversion is known. It computes the total return flow and depletion (consumption).

REQUIRED KNOWNS

☞ **TOTAL DIVERSION**

REQUIRED UNKNOWNS

☞ **TOTAL DEPLETION**

The **Solve given Diversion** dispatch method does the following:

If **Total Diversion Requested** is not specified, assume it is equal to **Total Diversion.** (The slot is not set, though).

If **Total Depletion Requested** is not specified, assume it is equal to the **Total Diversion Requested**, if valid. Otherwise assume it is equal to **Total Diversion**. (The slot is not set, though).

If **Total Diversion Requested** is zero, set **Total Return** flow equal to **Total Diversion** minus **Total Depletion Requested**.

Else, if **Total Diversion** is less than **Total Diversion Requested**, set

$$\text{Total Return Flow} = \text{Total Diversion} \times \left(1 - \frac{\text{Total Depletion Requested}}{\text{Total Diversion Requested}}\right)$$

Otherwise, **Total Diversion** is greater than **Total Diversion Requested**, set **Total Return Flow** equal to **Total Diversion** minus **Total Depletion Requested**.

Then, **Total Depletion** is set as the difference between **Total Diversion** and **Total Return Flow**.

Total Depletion Shortage is computed as **Total Depletion Requested** minus **Total Depletion**.

Total Diversion Shortage is computed as **Total Diversion Requested** minus **Total Diversion**.

21. Pumped Storage

Used to model a reservoir that can release water for power production and/or use a pumping system to increase storage.

The convergence algorithm used in the Pumped Storage Reservoir is detailed **HERE (Appendix A: Reservoir Convergence)**.

General Slots

☛ AVAILABLE PUMPS

Type: Table Series
Units: NONE VS. NONE
Description: pump unit vs. availability of that unit
Information: Availability is a positive value no greater than 1.0 indicating the decimal fraction of the capacity of the unit for the entire run time.
I/O: Required input
Links: Not linkable

☛ CONVERGENCE PERCENTAGE

Type: Table
Units: NONE
Description: A percentage value ranging from 0 to 1 used for convergence in all iterative calculations
Information: Click **HERE (Appendix A: Reservoir Convergence)** for more information on the convergence algorithm
I/O: Optional; defaults to 0.0001 if not input.
Links: Not linkable

☛ DIVERSION

Type: Series
Units: FLOW
Description: flow from the reservoir to a diverting object
Information: If not linked or input it is set to zero.
I/O: Optional; may be input or linked or neither
Links: May be linked to the Total Diversion slot on an Agg Diversion Site or the Total delivery Request slot on an AggDistribution Canal.

☛ Diversion Capacity
Type: Scalar Slot
Units: FLOW
Description: used to hold the maximum diversion physically possible from the reservoir
Information: This slot is used in the accounting system for allocation purposes and can be used in Rulebased Simulation
I/O: Input only
Links: Not linkable

☛ Elevation Volume Table
Type: Table
Units: LENGTH vs. VOLUME
Description: Reservoir Pool Elevation vs. Reservoir Storage
Information:
I/O: Required input
Links: Not linkable

☛ Energy
Type: Series Slot
Units: ENERGY
Description: product of the power generated by flow through the turbines and the length of the timestep
Information: This slot may also take the BEST_EFFICIENCY or MAX_CAPACITY flags, which allow Energy to act as input for dispatching, but solve for the value of energy assuming the generators are operating at best efficiency or maximum capacity.
I/O: Optional; if not input by the user, Energy is computed in the power calculations.
Links: Usually not linked

☛ Head vs Pump Flow
Type: Table
Units: LENGTH vs. FLOW
Description: operating head vs. maximum flow for each pump type
Information:
I/O: Required input
Links: Not linkable

☛ Head vs Pump Power
Type: Table
Units: LENGTH vs. POWER

Description: operating head vs. maximum pumping power for each pump type
Information:
I/O: Required input
Links: Not linkable

☛ Hours Utilized
Type: Table Series
Units: TIME, TIME
Description: hours of the timestep spent generating energy and pumping, respectively
Information:
I/O: Output only
Links: Not linkable

☛ Inflow
Type: Multi Slot
Units: FLOW
Description: inflow into the reservoir from upstream
Information:
I/O: Optional; if not input by the user, it is set through either mass balance computations or the propagation of values across the link.
Links: May be linked to one or more outflow slots of upstream objects.

☛ Max Iterations
Type: Table
Units: NOUNITS
Description: maximum number of allowable iterations for iterative loops in the solution algorithms
Information: Used in conjunction with Convergence Percentage as a stopping criterion for iterative calculations.
I/O: Optional; defaults to 100 if not input.
Links: Not linkable

☛ Operating Head
Type: Series Slot
Units: LENGTH
Description: elevation difference between the average Pool Elevation and the average Tailwater Elevation during a timestep
Information:
I/O: Output only
Links: Usually not linked

☛ Outflow

Type: Series Slot
Units: FLOW
Description: outflow from reservoir through the turbines
Information: Outflow is the sum of the Turbine Release and Spill
I/O: Optional; if not input by the user, it is set through mass balance computations.
Links: Linked to the Flow FROM Pumped Storage slot on another reservoir, or the Inflow slot on any object.

☛ Pool Elevation

Type: Series Slot
Units: LENGTH
Description: elevation of the water surface of the Reservoir
Information: There must be an initial value for either Storage or Pool Elevation given by the user for the first timestep.
I/O: Optional; if not input by the user, it is solved by the mass balance computations. It may take a TARGET flag indicated by the user for target operation solution.
Links: May be linked to Tailwater Elevation or Tailwater Base Value of an upstream object or to Elevation 1 or Elevation 2 of a Canal object.

☛ Power

Type: Series Slot
Units: POWER
Description: power generated by flow through the turbines
Information: Calculated by the power methods and cannot be input by the user.
I/O: Output only
Links: Not linkable

☛ Pump Capacity

Type: Series Slot
Units: POWER
Description: maximum pumping capacity
Information:
I/O: Output only
Links: Usually not linked

☛ Pump Energy

Type: Series Slot
Units: ENERGY
Description: energy required to run the pumps for the timestep

Information:
I/O: Can be input or solved for by the dispatch methods
Links: Usually not linked

☛ Pumped Flow
Type: Series
Units: FLOW
Description: flow into the reservoir by pumping
Information:
I/O: Optional; can be input or solved for
Links: Linked to the Flow TO Pumped Storage slot on another reservoir.

☛ Pump Power
Type: Series Slot
Units: POWER
Description: power required to run the pumps
Information:
I/O: Output only
Links: Usually not linked

☛ Pump Unit Types
Type: Table
Units: NONE vs. NONE
Description: pump unit number vs. type identification number of all pump units
Information:
I/O: Required input
Links: Not linkable

☛ Pumps Used
Type: Series
Units: NONE
Description: the number of pumps used during the timestep
Information: Specific pumps cannot be designated. Pumps will always be added in the best efficiency order.
I/O: Optional; input or output
Links: Usually not linked

☛ Return Flow
Type: Multi Slot
Units: FLOW

Description: flow returning from a diverting object
Information:
I/O: Optional; defaults to zero if not linked or input.
Links: May be linked to one or more Return Flow slots on Water User objects or the Total Return Flow slot on the Agg Diversion Site objects.

☛ Spill
Type: Series Slot
Units: FLOW
Description: sum of the regulated and unregulated spills and bypass
Information: May be input or solved for by **RiverWare**™ (see spill calculation methods).
I/O: Optional; may be input or solved for by **RiverWare**™ (see spill calculation methods).
Links: Usually not linked

☛ Storage
Type: Series Slot
Units: VOLUME
Description: volume of water stored in the reservoir
Information: May be flagged as a TARGET Storage value by the user. There must be an initial value for either Storage or Pool Elevation given by the user for the first timestep. If flagged as a TARGET, a target operation solution is used.
I/O: Optional; if not input by the user, it is set through mass balance computations.
Links: Usually not linked

☛ Tailwater Elevation
Type: Series
Units: LENGTH
Description: water surface elevation on the downstream side of the dam
Information: It can be linked to the Pool Elevation or Backwater Elevation of a downstream reservoir if the “Linked or Input” method is selected for the Tailwater category. Otherwise, it is calculated by the user method selected. It is used to compute the Operating Head used in the power calculations.
I/O: Optional; can be input, linked or calculated.
Links: It can be linked to the Pool Elevation or Backwater Elevation of a downstream reservoir if the “Linked or Input” method is selected for the Tailwater category.

☛ Total Inflows
Type: Series
Units: FLOW

Description: Summary slot displaying the flows into and out of the reservoir excluding the flows through the outlet works

Information: Total Inflows is calculated using the following equation:

Total Inflows = Inflow + Canal Flow +
Hydrologic Inflow + Hydrologic Inflow Adjust + Hydrologic Inflow Forecast +
Return Flow + Flow FROM Pumped Storage – Flow TO Pumped Storage – Diversion

Any component that is not in use or is not valid defaults to zero.

I/O: Output only

Links: Not linkable

☛ INFLOW SUM

Type: Series

Units: FLOW

Description: Sum of the total flows entering the reservoir at each timestep

Information: Inflow Sum is calculated using the following equation:

Inflow Sum = Inflow + Canal Flow +
Hydrologic Inflow + Hydrologic Inflow Adjust + Hydrologic Inflow Forecast +
Return Flow + Flow FROM Pumped Storage

I/O: Output only

Links: Not Linkable

☛ TURBINE RELEASE

Type: Series Slot

Units: FLOW

Description: flow through the turbines of a power reservoir (excluding spill)

Information:

I/O: Optional; solved for if not input.

Links: Usually not linked

21.1 User Methods

21.1.1 Power

The Power category user methods calculate the flow through the turbines (Turbine Release) and the Power and Energy generated. These methods require that the total Outflow of the Reservoir be known

21.1.1.1 None

This is the default method in the Power category. It contains no calculations for Power or Energy. There are no slots specifically associated with this method.

SLOTS SPECIFIC TO THIS METHOD

☛ **NONE**

The method first checks that Energy and Turbine Release are not input by the user. These slots cannot be input when None is the selected method. If either of these two slots are input, a **RiverWare**™ error will be posted and the simulation run aborted.

Next, the selected method in the Power Plant Failure category is executed. This sets the Power Plant Cap Fraction if necessary and checks for plant shutoff/failure. If the plant is shutoff/failed, the turbine Release is set to 0.0.

Otherwise, the Turbine Release is calculated as the difference between Outflow and Spill.

21.1.1.2 No Power Turbine Flow

The No Power Turbine Flow method is used to model Turbine Release without any power generation. Turbine Release is calculated as the Outflow minus Spill. The computed Turbine Release can not be larger than the Max Flow Through Turbines.

SLOTS SPECIFIC TO THIS METHOD

☛ **MAX FLOW THROUGH TURBINES**

Type: Table Slot
Units: LENGTH VS. FLOW
Description: relationship between Pool Elevation and Turbine Capacity
Information:
I/O: Required input
Links: Not linkable

The method first checks that Energy and Turbine Release are not input by the user. These slots are not valid for user input when the No Power Turbine Flow method is selected. If either of these two slots is input, a **RiverWare**™ error will be posted and the simulation run will be aborted. Pool Elevation is then used in an interpolation scheme to determine the maximum release from the Max Turbine Flow table.

Next, the selected method in the Power Plant Failure category is executed. This sets the Power Plant Cap Fraction if necessary and checks for plant shutoff/failure. If the plant is shutoff/failed, the turbine Release is set to 0.0.

Otherwise, the Turbine Release is set as either Outflow minus Spill or maximum release. It is set as the lesser of the two values because the Turbine Release must be less than the Turbine Capacity.

21.1.1.3 Plant Power Coefficient

The Plant Power Coefficient method calculates the Power and Energy generated based on the whole plant characteristics. If the Power Coefficient is specified, the Power is calculated directly, unless the **BEST EFFICIENCY** or **MAX CAPACITY** flag is set on Energy. If its not input, the Power Coefficient is found from the interpolation of the Best or Max Turbine Q and Power Coefficient tables using the current Operating Head. If the Turbine Release is less than the Best Turbine Q, the Best Power Coefficient Table is used. If the Turbine Release is greater than the Max Turbine Q, then the Max Power Coefficient Table is used. If the Turbine Release is between the two, an intermediate Power Coefficient Value is found by interpolation.

Slots Specific to this Method

☛ Best Hydro Capacity

Type: Series Slot
Units: POWER
Description: most efficient hydro capacity of the plant at the current timestep
Information: Solved iteratively based on Best Turbine Q and the Best Power Coefficient.
I/O: Output only
Links: Not linkable

☛ Best Power Coefficient

Type: Table Slot
Units: LENGTH vs. POWER per FLOW
Description: Operating Head vs. most efficient power coefficient
Information: The Power Coefficient relates turbine release to power generated. The Best Power Coefficient represents the most efficient power generation.
I/O: Required input

Links: Not linkable

☛ Best Turbine Q

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Operating Head vs. flow through turbine for most efficient power generation
Information:
I/O: Required input
Links: Not linkable

☛ Hydro Capacity

Type: Agg Series Slot
Units: POWER
Description: maximum hydro capacity of plant at the current timestep
Information: Solved iteratively based on Max Turbine Q and the Maximum Power Coefficient.
I/O: Output only
Links: Usually not linked

☛ Max Power Coefficient

Type: Table Slot
Units: LENGTH VS. POWER PER FLOW
Description: Operating Head vs. maximum power coefficient
Information: The Power Coefficient relates turbine release to power generated. The Max Power Coefficient represents the maximum Turbine Release.
I/O: Required input
Links: Not linkable

☛ Max Turbine Q

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Operating Head vs. maximum flow through the turbine
Information:
I/O: Required input
Links: Not linkable

☛ Minimum Power Elevation

Type: Table Slot
Units: LENGTH
Description: minimum Pool Elevation for power production

Information:
I/O: Required input
Links: Not linkable

☛ Plant Power Limit
Type: Series Slot
Units: POWER
Description: Power output is limited to this value
Information:
I/O: Optional; This constraint on power is only applied if the user inputs a value for the timestep.
Links: Not linkable

☛ Power Coefficient
Type: Series Slot
Units: POWER PER FLOW
Description: power generated per unit power release
Information:
I/O: Optional; can be input or calculated.
Links: Usually not linked

☛ Power Plant Cap Fraction
Type: Series Slot
Units: FRACTION
Description: the percentage of full capacity of the turbine units in the hydropower plant
Information:
I/O: Optional; The value of this slot defaults to 100% if not input by user.
Links: Not linkable

This method performs calculations to compute the power generated at each timestep.

First, Tailwater Elevation and Operating Head are determined based on the user method selected in the Tailwater category.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the Minimum Power Elevation was input by the user. If no value was input, a **RiverWare**™ error is posted and the simulation run is aborted. If the previous Pool Elevation is less than the Minimum Power Elevation or the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

7. Energy, Power, Hydro Capacity, and Best Hydro Capacity are set equal to zero.

8. If the Turbine Release is input or already set from the Dispatch Method "solveMB_givenInflowRelease," a **RiverWare**™ error is flagged and the run is aborted.

9. Turbine Release is set equal to zero.

Now, Operating Head is used to determine the maximum power release through interpolation on the Max Turbine Q table. The maximum power release value is multiplied by the Power Plant Cap Fraction to account for the state of the turbine units.

If Turbine Release is already set from the dispatch method "solveMB_givenInflowRelease", the following checks are performed:

- If Turbine Release is greater than Outflow - Spill, a **RiverWare**™ error is posted reading, "Requested Power Release is Greater than Outflow - Spill" and the run is aborted.
- If Turbine Release is greater than the maximum power release, a **RiverWare**™ error is posted reading, "Requested Turbine Release is greater than Maximum Turbine Capacity" and the run is aborted.

If the Turbine Release was input by the user, a **RiverWare**™ error is posted and the run is aborted. If neither the Energy nor the Turbine Release were input and the Energy was not set by a rules, the Turbine Release is set equal to the lesser of the Maximum Power Release or the value of the Outflow minus the Spill.

Using the calculated value of Operating Head, QmaxTemp and QbestTemp are obtained from the Max Turbine Q Table and the Best Turbine Q Table, respectively. Both values are then multiplied by the Power Plant Cap Fraction to obtain Qmax and Qbest. The Operating Head is also used to determine both the best power coefficient and the max power coefficient through interpolation of the Best Power Coefficient and Max Power Coefficient tables, respectively.

The following calculations are not performed if Energy is Input, set by a Rule, or flagged **BEST EFFICIENCY** or **MAX CAPACITY**. In these cases the Power, Energy, and Power Coefficient have already been calculated in Plant Power Coefficient Release.

If the Power Coefficient is not input by the user, the following steps are performed:

1. If the maximum power coefficient is greater than the best power coefficient, the following **RiverWare**™ error is posted, "best Power Coeff < full gate Power Coeff" and the simulation run is aborted.

2. If Qbest is greater than Qmax, the following **RiverWare**™ error is posted, "Best Turbine Q > Max Turbine Q" and the simulation run is aborted.

3. If Qbest equals Qmax, the Power Coefficient is set equal to the best power coefficient.

4. If none of the previous three conditions are satisfied and the Turbine Release is less than or equal to Qbest, the Power Coefficient is set equal to the best power coefficient.

5. If none of the previous four conditions are satisfied and the Turbine Release is less than Qmax, the Power Coefficient is calculated using the following equation:

$$\text{Power Coefficient} = \text{best power coefficient} + \frac{(\text{Turbine Release} - \text{Qbest})}{(\text{Qmax} - \text{Qbest})} \times (\text{max power coefficient} - \text{best power coefficient})$$

6. If none of the previous four conditions are true, the Power Coefficient is set equal to the max power coefficient.

Power is then calculated using the following equation:

$$\text{Power} = \text{Power Coefficient} \times \text{Turbine Release}$$

If the user has input the Plant Power Limit, the following steps are taken:

1. If the Power Coefficient is input by the user, Power and Turbine Release may need to be recalculated. If the Power is greater than the Plant Power Limit, Power is set equal to the Plant Power Limit and Turbine Release is recalculated as the Plant Power Limit divided by the Power Coefficient.

2. If the Power Coefficient is not input and the Plant Power Limit is exceeded; the Turbine Release, Power, and Power Coefficient may need to be recalculated. If the Power Coefficient is equal to the best power coefficient, the plant is already operating at best efficiency. Therefore, the Turbine Release is set equal to the Plant Power Limit divided by the Power Coefficient and the rest of the flow is spilled. The Power and Power Coefficient do not need to be recalculated.

If the Power Coefficient is not equal to the best power coefficient, Turbine Release, Power, and the Power Coefficient need to be recalculated. This is done through the following steps:

- Temporary variables are calculated from the following equations:

$$\text{power at best} = \text{best power coefficient} \times \text{Qbest}$$

$$\text{power at max} = \text{max power coefficient} \times \text{Qmax}$$

$$\text{Qlimit} = \text{Qbest} + \frac{\text{Plant Power Limit} - \text{power at best}}{(\text{power at max} - \text{power at best}) \times (\text{Qmax} - \text{Qbest})}$$

$$\text{PClimit} = \text{best power coefficient} + \frac{\text{Plant Power Limit} - \text{power at best}}{(\text{power at max} - \text{power at best}) \times (\text{max power coefficient} - \text{best power coefficient})}$$

- If Qlimit is greater than Qmax: the Power Coefficient is set equal to the max power coefficient, Turbine release is set equal to Qmax, and Power is set equal to the power at max.
- If Qlimit is less than Qbest: the Power Coefficient is set equal to the best power coefficient, Turbine Release is set equal to Qlimit, and Power is set equal to the Plant Power Limit.
- If Qlimit is less than Qmax but greater than Qbest: the Power Coefficient is set to equal to PClimit, Turbine Release is set equal to Qlimit, and Power is set equal to the Plant Power Limit.

If the Spilled Energy Power Coefficient is visible on the reservoir and it is not input, it is set equal to the Power Coefficient.

Energy is calculated as Power multiplied by the timestep (in hours).

The following calculations take always take place, regardless of the flag on Energy:

If either the Turbine Release is equal to the maximum power release or the Energy is at the maximum capacity, Hydro Capacity is set equal to Power and the Best Hydro Capacity is obtained from the getHydroCap function. If the energy is at the **Best Efficiency**, the Best Hydro Capacity is set equal to the Power and the Hydro Capacity is obtained from the getHydroCapacity function. If neither the Turbine Release is equal to the maximum power release, the Energy is at the maximum capacity, nor the Energy is at the **Best Efficiency**, both Hydro Capacity and Best Hydro capacity are obtained with the getHydroCap function.

21.1.1.4 Plant Efficiency Curve

The Plant Efficiency Curve method calculates the Power and Energy generated based on the whole plant characteristics. If the Power Coefficient is specified, the Power is calculated directly, unless the **BEST EFFICIENCY** or **MAX CAPACITY** flag is set on Energy. If the Power Coefficient is not input, the Power is found by a 3-D interpolation of the Plant Power Table using the current, average Operating Head and Turbine Release. The Power Coefficient is calculated as Power divided by Turbine Release.

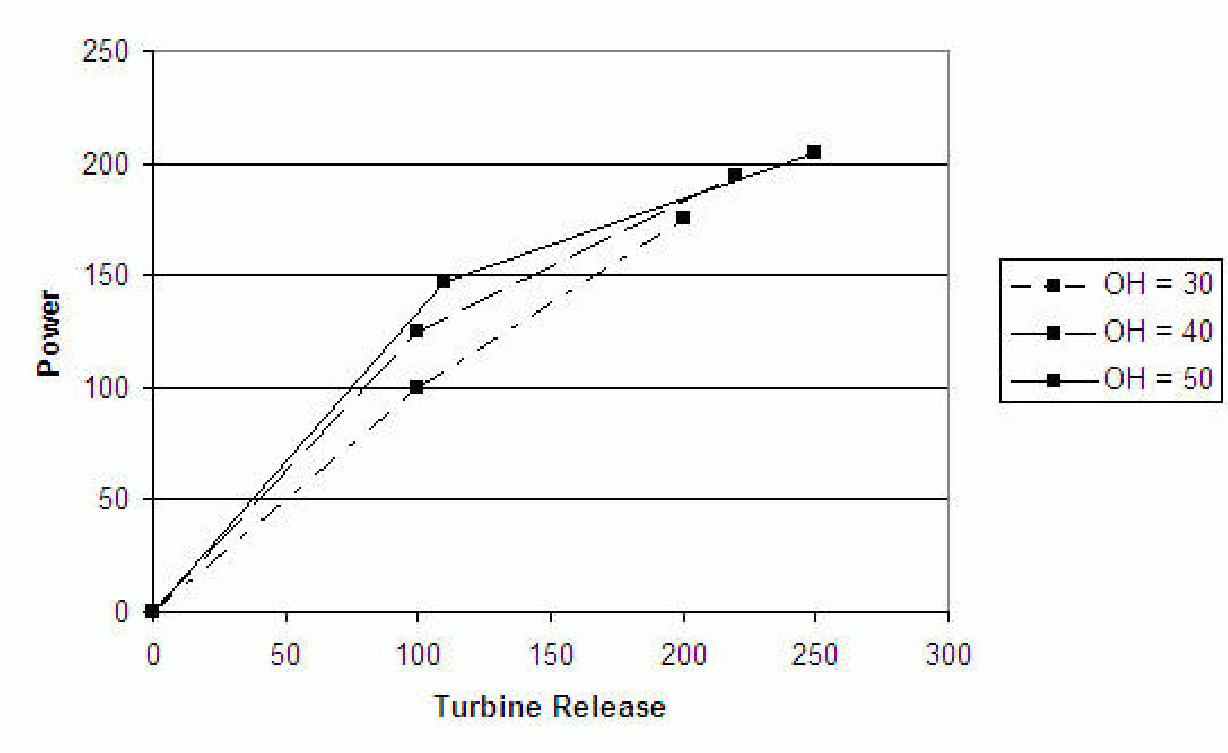
Slots Specific to this Method

☛ Plant Power Table

Type: Table Slot
Units: LENGTH, FLOW, POWER
Description: 3-D table used to determine power using interpolation

Information: Data must be entered into the table in increasing, concave blocks of the same Operating Head for the 3-dimensional table interpolation to work correctly. For every block of the same Operating Head in column 1, Turbine Release should be listed in increasing, concave order in column 2, and the corresponding Power in column 3. There should also be a point of zero Turbine Release and zero Power for each operating head. The second to last row for each operating head is the point of best efficiency. The last row for each operating head is the point of maximum Turbine Release and maximum Power production. If there are only two rows for a given operating head, both the **best efficiency** and **max capacity** are equal to the second row. The table shown below is an example of the proper way to formulate the Plant Power Table. The graph displays the increasing concave blocks at each operating head.

Operating Head	Turbine Release	Power
30	0	0
30	100	100
30	200	175
40	0	0
40	100	125
40	220	195
50	0	0
50	110	147
50	250	205



I/O: Input Only

Links: Not Linkable

☛ Power Coefficient

Type: Series Slot
Units: POWER PER FLOW
Description: power generated per unit flow release
I/O: Optional; if input, it is used to compute power. Otherwise, power is computed from the Plant Power Table
Links: Not usually linked

☛ Hydro Capacity

Type: Agg Series Slot
Units: POWER
Description: This is the maximum power that can be produced at the current timestep
Information: Solved for iteratively based on the Operating Head and maximum possible release.

I/O: Output Only

Links: Not Linkable

☛ Best Hydro Capacity

Type: Series Slot
Units: POWER
Description: This is the power that would be produced at the most efficient operating point at the current timestep.
Information: Solved for iteratively based on the most efficient operating point and the corresponding release.
I/O: Output Only

Links: Not Linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: FRACTION
Description: Must be a number less than or equal to 1. If not input, automatically set to 1.
Information: This is the percentage of full capacity of the turbine units in the hydropower plant. For example, if only half of the turbines are operational (and they are all the same), this value would be 0.5
I/O: Can be input by user. If not, value is set to 1.

Links: Not Linkable

☛ Minimum Power Elevation

Type: Series Slot
Units: LENGTH
Description: The minimum elevation at which the reservoir can still produce power.
I/O: Input Only

Links: Not Linkable

☛ Plant Power Limit

Type: Series Slot
Units: POWER
Description: The max power that the plant can produce at a given timestep.
I/O: Optional, only applies if input by user

Links: Not Linkable

☛ Power Curvature Tolerance

Type: Scalar
Units: NONE

Description: The power curvature tolerance is used to account for anomalies in Plant Power Table data and round off error while calculating slopes.
Information: Although the units for the slot are “None”, the comparison is implicitly using (MW/cms).
I/O: Input or defaults to $1X10^{-6}$
Links: Not linkable

At the start of an optimization run, the Plant Power Table is checked for concavity. The slope of each segment for each block is calculated as:

$$Slope_{Segment} = \frac{Power_i - Power_{i-1}}{TurbineRelease_i - TurbineRelease_{i-1}}$$

The table is considered concave if:

$$Slope_{Segment} \le Slope_{PreviousSegment} + PowerCurvatureTolerance$$

The table is required to be concave for optimization runs, but not for simulation or rulebased simulation runs.

This method performs calculations to compute the power generated at each timestep.

First, Tailwater Elevation and Operating Head are determined based on the user method selected in the Tailwater category.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the Minimum Power Elevation was input by the user. If no value was input, a **RiverWare**™ error is posted and the simulation run is aborted. If the previous Pool Elevation is less than the Minimum Power Elevation or the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Energy, Power, Hydro Capacity, and Best Hydro Capacity are set equal to zero.
2. If the Turbine Release is input or already set from the Dispatch Method “solveMB_givenInflowRelease,” a **RiverWare**™ error is flagged and the run is aborted.
3. Turbine Release is set equal to zero.

Operating Head is used to determine the maximum power release through interpolation on the Plant Power Table. The maximum power release value is multiplied by the Power Plant Cap Fraction to account for the state of the turbine units.

If Turbine Release is already set from the dispatch method "solveMB_givenInflowRelease", the following checks are performed:

- If Turbine Release is greater than Outflow - Spill, a **RiverWare**™ error is posted reading, "Requested Power Release is Greater than Outflow - Spill" and the run is aborted.
- If Turbine Release is greater than the maximum power release, a **RiverWare**™ error is posted reading, "Requested Turbine Release is greater than Maximum Turbine Capacity" and the run is aborted.

If the Turbine Release was input by the user, a **RiverWare**™ error is posted and the run is aborted. If neither the Energy nor the Turbine Release were input and the Energy was not set by a rules, the Turbine Release is set equal to the lesser of the Maximum Power Release or the value of the Outflow minus the Spill.

The following calculations are not performed if Energy is Input, set by a Rule, or flagged **BEST EFFICIENCY** or **MAX CAPACITY**. In these cases the Power and Energy have already been calculated in Plant Efficiency Curve Release.

If the Power Coefficient is input by the user,

$$Power = TurbineRelease \times PowerCoefficient$$

Otherwise, Power is found directly from the Plant Power Table using the current Operating Head and the Turbine Release from above. The power coefficient is now calculated as:

$$PowerCoefficient = Power / TurbineRelease$$

If the user has input the Plant Power Limit, the following steps are taken:

1. If the Power Coefficient is input by the user, Power and Turbine Release may need to be recalculated. If the Power is greater than the Plant Power Limit, Power is set equal to the Plant Power Limit and Turbine Release is recalculated as the Plant Power Limit divided by the Power Coefficient.
2. If the Power Coefficient is not input and the Plant Power Limit is exceeded; the Turbine Release, Power, and Power Coefficient need to be recalculated. The Power is set equal to the Plant Power Limit and the Turbine Release is found using 3-D interpolation of the Plant Power Table. The Power Coefficient is then calculated as Power divided by Turbine Release.

Energy is then calculated as Power multiplied by the timestep length.

The following calculations take always take place, regardless of the flag on Energy:

If either the Turbine Release is equal to the maximum power release or the Energy is at the maximum capacity, Hydro Capacity is set equal to Power and the Best Hydro Capacity is computed iteratively. If the energy is at the **Best Efficiency**, the Best Hydro Capacity is set equal to the Power and the Hydro Capacity is computed iteratively. If the Turbine Release is not equal to the maximum power release, the Energy is not at the maximum capacity, and the Energy is not at the **Best Efficiency**, both Hydro Capacity and Best Hydro capacity are computed by an iterative algorithm.

Notes on Power Plant Cap Fraction

If the Power Plant Cap Fraction is input by the user, it is necessary for the Plant Power Table to basically be scaled back to account for the operating points when the turbines are operating at less than 100%. To do this, when Turbine Release is known and Power is to be found using the Plant Power Curve, Turbine Release is divided by the Power Plant Cap Fraction. This point is then found in the Plant Power Curve for the current operating head and the Power is found using 3-D interpolation. Finally the Power is multiplied by the Power Plant Cap Fraction to get the actual Power produced for the current timestep.

If Power is known, and Turbine release is to be found in the table. Power is multiplied by the Power Plant Cap Fraction and then this point is found in the Plant Power Curve to solve for Turbine Release. Turbine Release is then divided by the Power Plant Cap Fraction to get the actual Turbine Release for the current timestep.

21.1.1.5 Plant Power Equation

The Plant Power Equation method is used to calculate Power and Energy using the water power equation.

Slots Specific to this Method

☛ Head Loss

Type: Table Slot
Units: LENGTH
Description: The head loss water incurs before it reaches the turbines.
Information: The slot is set to zero if not input by the user.
I/O: optional
Links: Not linkable

☛ Minimum Power Elevation

Type: Table Slot
Units: LENGTH
Description: Minimum pool elevation at which power can be generated
Information: Single value in a 1x1 table slot
I/O: Required input
Links: Not linkable

☛ Net Head vs Max Turbine Release

Type: Table Slot
Units: LENGTH VS. FLOW
Description: relationship between the net head and the maximum possible turbine release
Information: Net Head must account for any head loss
I/O: Required input
Links: Not linkable

☛ Net Head vs Plant Efficiency

Type: Table Slot
Units: Length vs None
Description: This table allows you to specify the efficiency as a function of (previous) Net Head.
Information: This table is used only when the Plant Efficiency Value is empty. The Net Head used in this table look up comes from the previous timestep's operating head.
I/O: Optional Input
Links: Not Linkable

☛ Plant Efficiency Value

Type: Table Slot
Units: NONE
Description: the decimal percent efficiency at which the plant is operating
Information: Single value in a 1x1 table slot. Plant efficiency should incorporate both generator efficiency and turbine efficiency.
I/O: Optional Input, if specified, it must be between 0 and 1.
Links: Not linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: NONE
Description: decimal fraction of the power capacity at which the plant is operating
Information: Used in the case of outages or reductions in the plant operating capacity.
I/O: Defaults to 1.0 if not input.
Links: Not linkable

☛ Plant Power Limit

Type: Series Slot
Units: Power
Description: The user specified upper limit on power production

Information: If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit and the Energy is recalculated. A new Turbine Release are then calculated based on the Plant Power Limit.
I/O: Optional Input or set by a rule.

The method first checks whether Energy is either user input or set by rules. If it is, the method finishes successfully and exits-- all power calculations were already performed in the Plant Power Equation Release method.

Otherwise, Tailwater Elevation and Operating Head are determined based on the user method selected in the Tailwater category.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the Minimum Power Elevation was input by the user. If no value was input, a **RiverWare**™ error is posted and the simulation run is aborted. If the previous Pool Elevation is less than the Minimum Power Elevation or the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Energy and Power are set equal to zero.
2. If the Turbine Release is input or already set from the Dispatch Method “solveMB_givenInflowRelease,” a **RiverWare**™ error is flagged and the run is aborted.
3. Turbine Release is set equal to zero.

Once the initial checks are performed, the method calculates Net Head and Turbine Release as:

$$NetHead = OperatingHead - HeadLoss$$

Next, the Max Turbine Release for the current Net Head is interpolated from the Net Head vs. Max Turb Release table.

If there is a valid value in the Plant Efficiency Value slot, that is used for the *efficiency.* Otherwise, given the net head from the previous timestep (Operating Head at previous timestep minus Head Loss), the *efficiency* is interpolated from the Net Head vs Efficiency table. The previous Operating Head is used as an approximation so as not to introduce an additional variable in the iteration. As a result, the Tailwater Elevation at the initial timestep must be input. The net head for the initial timestep is the initial Pool Elevation minus the initial Tailwater Elevation minus Head Loss.

The method checks whether Turbine Release is user input or set by a link or a rule. If Turbine Release is known at the dispatch level, the method will check that it is not greater than Outflow minus Spill or the Max Turbine Release given the current Net Head. If either of these are true, an error will be posted and the run will abort. Otherwise, the known Turbine Release value will be used in the Power calculations. If Turbine Release is **not** user input or solved for in the dispatch methods, it is calculated as the minimum of the Max Turbine Release (given the current Net Head) and Outflow minus Spill (either unregulated spill or user specified regulated spill):

$$TurbineRelease = Min(MaxTurbineRelease \times PowerPlantCapFraction, Outflow - Spill)$$

Note: If Turbine Release is set to MaxTurbineRelease, it means there is still some remaining water that must be passed via regulated spill. This will be calculated in the spill calculations. If Turbine Release is set to Outflow minus Spill, it means that Spill consists of Unregulated Spill and any input Regulated Spill-- all other water will pass through the turbines.

Once efficiency, Net Head, and Turbine Release are all known, Power is solved for using the Power Equation:

$$Power = \frac{\text{Turbine Release} \times \text{Net Head} \times \text{efficiency}}{\text{Unit Compatibility Factor}}$$

The unit compatibility factor comes from balancing units and is 102.01697767 in internal RiverWare units.

If the computed Power is greater than the Plant Power Limit, the Power is reset to the Plan Power Limit and Turbine Release is recomputed by solving the above equation for Turbine Release.

Lastly, Energy is computed as Power multiplied by the time length of the timestep.

$$Energy = \text{Power} \times \text{Length of Timestep}$$

21.1.1.6 Unit Generator Power

The Unit Generator Power method is used to calculate Power and Energy for generating units with individual characteristics. The generating units are grouped by unit type for ease of data entry.

Slots Specific to this Method

☛ Best Generator Flow

Type: Table Slot
Units: LENGTH VS. FLOW
Description: a table for each unit type which gives the relationship between operating head and flow through the generator when operating at best efficiency

Information: There must be a block of data for each unit type given in the Generator Unit Types table. The table is representative of a single unit within the specified unit type.
I/O: Required input
Links: Not linkable

☛ Best Generator Power

Type: Table Slot
Units: LENGTH VS. POWER
Description: a table for each unit type which gives the relationship between operating head and power produced by the generator when operating at best efficiency
Information: There must be a block of data for each unit type given in the Generator Unit Types table. The table is representative of a single unit within the specified unit type.
I/O: Required input
Links: Not linkable

☛ Full Generator Flow

Type: Table Slot
Units: LENGTH VS. FLOW
Description: a table for each unit type which gives the relationship between operating head and flow through the generator when operating at full capacity
Information: There must be a block of data for each unit type given in the Generator Unit Types table. The table is representative of a single unit within the specified unit type.
I/O: Required input
Links: Not linkable

☛ Full Generator Power

Type: Table Slot
Units: LENGTH VS. POWER
Description: a table for each unit type which gives the relationship between operating head and power produced by the generator when operating at full capacity
Information: There must be a block of data for each unit type given in the Generator Unit Types table. The table is representative of a single unit within the specified unit type.
I/O: Required input
Links: Not linkable

☛ Generator Unit Types

Type: Table Slot

Units: NONE
Description: a list of each generating unit and the corresponding unit type
Information: More than one generating unit can be assigned to a given unit type. The unit type must be an integer value beginning with 1 and increasing by increments of 1.
I/O: Required input
Links: Not linkable

☛ Generators Available and Limit

Type: Table Series Slot
Units: FRACTION AND POWER
Description: a time series specifying the availability and power limit of each generating unit.
Information: Availability is a number between 0 and 1 which represents the percentage of the timestep that the unit is available. There must be a block of data for each row in the Generator Unit Types table. The Power Limit has no effect on the flow through the turbines.
I/O: Required input
Links: Not linkable

☛ Hydro Capacity

Type: Agg Series Slot
Units: POWER
Description: the maximum power production possible at the current timestep
Information: This value is the sum of all generators operating at full capacity for the given operating head at the current timestep.
I/O: Output only
Links: Could be linked to a Data Object, but usually not linked.

☛ Minimum Power Elevation

Type: Table Slot
Units: LENGTH
Description: the minimum pool elevation required for power production
Information: When the Pool Elevation drops below this value, a warning is posted and no power is produced.
I/O: Required input
Links: Not linkable

☛ Power Coefficient

Type: Series Slot
Units: POWER/FLOW

Description: power generated per unit power release
Information: This coefficient corresponds to the efficiency of the entire plant. It is not used in calculation and is displayed only for the benefit of the user.
I/O: Output only
Links: Could be linked to a Data Object, but usually not linked.

The Unit Generator Power method begins by computing the availability and power limits of each unit type. Availability and power limit values are computed as the sum of the values from the availability and power limit columns, respectively, in the Generators Available and Limit slot. A value for availability and power limit is computed for each unit type.

First, Tailwater Elevation and Operating Head are determined based on the user method selected in the Tailwater category.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction if necessary (the default is 1.0) and checks for plant shutoff/ failure.

If the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Energy and Power are set equal to zero.
2. If the Turbine Release is input or already set from the Dispatch Method "solveMB_givenInflowRelease," a **RiverWare**™ error is flagged and the run is aborted.
3. Turbine Release is set equal to zero.

Then the efficiency of each unit type is calculated by the following equation:

$$efficiency = \frac{powerTemp}{flowTemp}$$

PowerTemp and flowTemp, both local variables, are computed from the Best Generator Power and Best Generator Flow tables, respectively, using the current Operating Head. Each unit type is then sorted in descending order based on the computed efficiency.

Turbine Release has already been computed by the dispatch method and the power produced from the known Turbine Release must be calculated. The method begins to add entire unit types (operating according to the best flow and power tables and beginning with the most efficient type) until the Turbine Release is exceed or all the unit types have been added. If the power generated by a particular unit type exceeds the power limit for that unit type, the power produced from that type is set to the power limit. The power limit has no effect on the flow going through the turbines. If the Turbine Release is exceeded, the last generator type is interpolated to compute the Power exactly (see equation below). However, if all the unit types have been added and the Turbine Release cannot be met, the method assumes all unit types are operating at full capacity (according to the Full Generator Flow and Full Generator Power tables). Then if the Turbine Release is exceeded, the last generator type added is interpolated to compute the Power exactly (see equation below). However, if the Turbine Release still cannot be met, all unit types are run at full capacity. Turbine Release is reset to the maximum flow through the turbines and Power is set as the maximum power produced by the turbines (at the given operating head). The spill must be recalculated to handle the excess Turbine Release that could not be met.

The interpolation equation used to calculate Power is given below:

$$\text{Power} = \text{oneLessTypePower} + \frac{\text{Turbine Release} - \text{oneLessTypeFlow}}{\text{cumulativeFlow} - \text{oneLessTypeFlow}} \bullet (\text{cumulativePower} - \text{oneLessTypePower})$$

where oneLessTypePower is the power produced from all the previous types added (excluding the most recent type added); oneLessTypeFlow is the flow through all the previous unit types (excluding the most recent type added); cumulativePower is the power produced from all the unit types added (including the most recent type); and cumulativeFlow is the flow through all the unit types added (including the most recent type).

Note: The above equation assumes the relationship between power and flow is linear, regardless of the actual relationship specified in the power and flow tables. It is also interpolating over an entire type of generators.

The Power Coefficient is then calculated by the following equation:

$$\text{Power Coefficient} = \frac{\text{Power}}{\text{Turbine Release}}$$

If all the unit types were added, the Spilled Energy Power Coefficient is equal to the Power Coefficient. However, if all the types were not added, the Spilled Energy Power Coefficient is set equal to the efficiency of the last type added.

Energy is calculated as the product of the Power and the timestep length. Hydro Capacity is set as the power produced from all units operating at full capacity.

21.1.1.7 Peak and Base

The Peak and Base method computes the Power and Energy generated by the entire plant based on the fraction of each timestep operated at peak flow and base flow. It is a long timestep method, modeled after the U. S. Bureau of Reclamation’s CRSS peak-base power calculation. A peaking flow value is first determined from the Outflow, Tailwater Elevation and Best Generator Flow. A minimum Base Flow and Power production are assumed for the entire timestep. Next, the number of hours to operate at peak power is calculated from the remaining volume of water released during that timestep. Peak production and base production are then added to determine the total Energy. Power is calculated by dividing the Energy by the timestep length in hours. Power Capacity is the power that could be generated if the flow is directed through the turbine(s) given an operating head. This is added to distinguish between actual power production and the power that could be produced.

Slots Specific to this Method

☛ Base Flow

Type: Series Slot
Units: FLOW
Description: minimum flow through turbines to produce energy
Information: This value is read from the Base Flow Table
I/O: Output only
Links: Not linkable

☛ Base Flow Table

Type: Table Slot
Units: FLOW vs. FLOW
Description: Outflow from the Reservoir vs. base flow
Information: This table gives the minimum flow required through the turbines as a function of the average total outflow from the Reservoir.
I/O: Required input
Links: Not linkable

☛ Best Generator Flow

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Operating Head vs. flow through the turbine at best efficiency
Information: The minimum and maximum values of operating head in this table are used as limiting values. The operating head is reset to the min. or max if it exceeds these constraints.
I/O: Required input
Links: Not linkable

☛ Best Generator Power

Type: Table Slot
Units: LENGTH vs. POWER
Description: Operating Head vs. power at best efficiency
Information: power produced by the entire plant at base energy flow
I/O: Required input

Links: Not linkable

☛ Maximum Turbine Power

Type: Table Slot
Units: POWER
Description: maximum turbine power output
Information:
I/O: Required input
Links: Not linkable

☛ Min and Max Operating Head

Type: Table Slot
Units: LENGTH
Description: the minimum and maximum operating head for the turbines
Information:
I/O: Required input
Links: Not linkable

☛ Number of Units

Type: Table Slot
Units: NONE
Description: integer number of turbines in plant
Information:
I/O: Required input
Links: Not linkable

☛ Off Peak Capacity

Type: Table Slot
Units: POWER
Description:
Information:
I/O: Required input
Links: Not linkable

☛ Peak Flow

Type: Series Slot
Units: FLOW
Description: most efficient flow through turbines for the current Operating Head
Information:
I/O: Output only

Links: Not linkable

☛ Peak Hours

Type: Series Slot
Units: TIME
Description: the number of hours operated at peak flow
Information:
I/O: Output only
Links: Not linkable

☛ Plant Efficiency

Type: Series Slot
Units: FRACTION
Description: ratio of actual power produced to peak and base theoretical power
Information:
I/O: Output only
Links: Not linkable

☛ Power Capacity

Type: Agg Series Slot
Units: POWER
Description: power that could be produced if flow is directed through the turbines given the operating head.
Information: Calculated by the two peak power methods and cannot be input by the user.
I/O: Output only
Links: Not linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: FRACTION
Description: the percentage of full capacity of the turbine units in the hydropower plant
Information:
I/O: Optional; the value of this slot defaults to 100% if not input by user
Links: Not linkable

First, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction if necessary (the default is 1.0) and checks for plant shutoff/failure.

If the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Energy, Power, Base Flow, Peak Flow, Peak Power, Power Capacity, and Plant Efficiency are set equal to zero.

2. If the Turbine Release is input or already set from the Dispatch Method "solveMB_givenInflowRelease," a **RiverWare**™ error is flagged and the run is aborted.

3. Turbine Release is set equal to zero.

4. Tailwater and Operating Head are computed and set. No further computations are performed.

If either the Energy or Turbine Release is input by the user, a **RiverWare**™ error is posted and the simulation run is aborted. These are not valid input slots for the Peak and Base method. Peak Flow and Peak Power are then calculated as follows:

1. If either the Maximum Turbine Power is not valid or it is less than 0.00000001 MW, a **RiverWare**™ Error is flagged and the simulation run is aborted.

2. The Maximum Turbine Power is then used with the Best Generator Power table to obtain a value for the local variable, headAtMaxPower.

3. headAtMaxPower is used with the Best Generator Flow table to obtain a value for the local variable, flowAtMaxPower. The local variable, flow is temporarily set as flowAtMaxPower.

4. The local variable, efficiencyAtMaxPower is calculated by the following formula:

$$\text{efficiencyAtMaxPower} = \text{Maximum Turbine Power} \times \frac{1000000}{\text{flowAtMaxPower} \times \text{headAtMaxPower} \times 999.99 \times 9.79908}$$

where 999.99 is the density of water (Kg/M^3) at five degrees C and 9.79908 is gravitational acceleration (M/s^2) at 37 degrees North latitude.

5. **RiverWare**™ then iterates while:

- The absolute difference between Qnew and flow is greater than 5 cfs.
- The number of iterations is less than the maximum number of iterations.
- The Operating Head is greater than the Minimum Operating Head.

The following calculations and evaluations are inside the iterative loop:

- The local variable, Qnew is set equal to flow
- The local variable, plantFlow is determined using the following equation:

$$\text{plantFlow} = \text{Qnew} \times \text{Number of Units} \times \text{Power Plant Cap Fraction}$$

- Flow is set equal to plantFlow
- The user selected Tailwater calculation is performed
- The Operating Head is calculated
- If the Operating Head is greater than the Maximum Operating Head, Operating Head is set equal to the maximum Operating Head
- If the Operating Head is less than the Minimum Operating Head, Operating Head is set equal to the Minimum Operating Head
- If the Operating Head is greater than the headAtMaxPower, the flow is calculated using the following equation:

$$\text{flow} = \text{Maximum Turbine Power} \times \frac{1000000}{\text{Operating Head} \times \text{efficiencyAtMaxPower} \times 999.99 \times 9.79908}$$

- If the Operating Head is less than the Maximum Operating Head, greater than the Minimum Operating Head, and less than the headAtMaxPower, flow is obtained from the Best Generator Flow table and the Operating Head.

This set of calculations is repeated until the iteration criteria described above are met.

6. If Operating head is less than the Minimum Operating Head; Turbine Release, Energy, Power, Power Capacity, Peak Flow, Peak Hours, and flow are all set to zero. Then the Tailwater method is re-executed. If Operating Head is greater than the minimum Operating Head and the headAtMaxPower, Peak Power is set equal to Maximum Turbine Power. If the Operating Head is greater than the Minimum Operating Head and less than the headAtMaxPower, Peak Power is determined from the Best Generator Power table using Operating Head.

Once Peak Power and Peak Flow (called “flow” in calculations described above) are calculated, Base Power and Base Flow can be determined. Base Power is set equal to Off Peak Capacity and Base Flow is determined from Outflow and the Base Flow Table.

If Outflow minus Unregulated Spill is greater than the product of Peak Flow, Number of Units, and Power Plant Cap Fraction; the following steps are taken:

1. The Tailwater method selected by the user is executed.
2. The Operating Head is calculated.
3. The local variable, headAtMaxPower, is obtained from the Best Generator Power table using the Maximum Turbine Power.

4. The local variable, flowAtMaxPower, is obtained from the Best Generator Flow table using the headAtMaxPower.

5. The local variable, efficiencyAtMaxPower, is computed using the following formula:

$$\text{efficiencyAtMaxPower} = \frac{\text{Maximum Turbine Power} \times 1000000}{(\text{flowAtMaxPower} \times \text{headAtMaxPower} \times 999.99 \times 9.79908)}$$

6. If Operating Head is greater than headAtMaxPower, Peak Flow and Peak Power are calculated using the following equations:

$$\text{Peak Flow} = \text{Maximum Turbine Power}$$
$$\times \frac{1000000}{\text{Operating Head} \times \text{EfficiencyAtMaxPower} \times 999.99 \times 9.79908}$$

$$\text{Peak Power} = \text{Maximum Turbine Power}$$

7. If the Operating Head is less than or equal to the headAtMaxPower, Peak Flow and Peak Power are determined using the Operating Head in conjunction with the Best Generator Flow and Best Generator Power tables, respectively.

The Peak Flow slot represents the flow through the entire power plant. Therefore, the value in this slot is calculated as Peak Flow times Number of Units times Power Plant Cap Fraction. In the calculations that follow, Peak Flow represents the slot value just calculated. Plant Peak Power is calculated as Peak Power times Number of Units times Power Plant Cap Fraction.

The number of hours required to operate at base and peak flows are computed next using the following equations:

$$\text{Peak Flow Volume} = (\text{Outflow} - \text{Spill} - \text{Base Flow}) \times \text{timestep (in seconds)}$$

$$\text{Peak Hours} = \frac{\text{Peak Flow Volume}}{(\text{Peak Flow} - \text{Base Flow}) \times 3600}$$

$$\text{Base Hours} = \frac{\text{Timestep (in seconds)}}{3600} - \text{Peak Hours}$$

If Peak Hours is greater than the length of the timestep; Peak Hours is set equal to the timestep, Base Hours are set to zero, Turbine Release is set to Peak Flow, and Total Controlled Release is calculated as Outflow minus Unregulated Spill. If Peak Hours is less than or equal to the length of the timestep, Peak Hours and Base Hours remain as calculated by the above formulas, Turbine Release is Outflow minus Spill, and Total Controlled Release is set equal to the Peak Flow.

The theoretical and actual energy production are computed next. The local variable, peakEnergy is calculated as Peak Hours times Plant Peak Power. The local variable, baseEnergy is calculated as Base Hours times Base Power.

The local variable, bestBaseTheor (representing the theoretical, most efficient base energy) is calculated using the following equation:

$$\text{bestBaseTheor} = \text{Base Flow} \times \text{Operating Head at Base Flow} \times \text{Base Hours} \times 0.00980229$$

The local variable bestPeakTheor (representing the theoretical most efficient peak energy) is calculated using the following equation:

$$\text{bestPeakTheor} = \text{Peak Flow} \times \text{Operating Head at Peak Flow} \times \text{Peak Hours} \times 0.00980229$$

The value, 0.00980229, is a conversion factor necessary for energy to have units of megawatt-hours.

$$\text{Energy (MWH)} = 62.4\ (\text{lb/ft3}) \times Q\ (\text{cfs}) \times \text{time (hrs)} \times \text{Head (ft)} \times 1\ \text{KW} / 737.56\ (\text{ft-lb/s}) \times 1\ \text{MW} / 1000\ \text{KW}$$

For Q in cms and Head in meters the final conversion is 0.00980229.

Finally, the following slots are set:

$$\text{Power Capacity} = \text{Plant Peak Power}$$

$$\text{Energy} = \text{peakEnergy} + \text{baseEnergy}$$

$$\text{Power} = \frac{\text{Energy}}{\text{timestep length (hours)}}$$

$$\text{Plant Efficiency} = \frac{\text{Energy}}{\text{bestPeakTheor} + \text{bestBaseTheor}}$$

If a spill method is selected which utilizes the Spilled Energy Power Coefficient and this value is not input by the user, it is set as:

$$\text{Spilled Energy Power Coefficient} = \frac{\text{Power}}{\text{Turbine Release}}$$

If Turbine Release is zero, the Spilled Energy Power Coefficient is also zero.

21.1.1.8 Peak Power

The Peak Power method is similar to the Peak and Base method except that it computes power and energy based on Peak Flow only. A peaking flow value is first determined from the Outflow, Tailwater Elevation, and Best Generator Flow. The number of hours to operate at peak power is then calculated from the volume of water released during that timestep. A distinction is made between actual power production and the power that could be produced. Power Capacity is the peak power capacity. Power is calculated by dividing the energy by the timestep length in hours. There is no Base Flow power production.

Slots Specific to this Method

☛ Best Generator Flow

Type: Table Slot
Units: LENGTH vs. FLOW
Description: operating head vs. flow through the turbine at best efficiency
Information: The minimum and maximum values of operating head in this table are used as limiting values. The operating head is reset to the min. or max if it exceeds these constraints.
I/O: Required input
Links: Not linkable

☛ Best Generator Power

Type: Table Slot
Units: LENGTH vs. POWER
Description: operating head vs. power at best efficiency
Information:
I/O: Required input
Links: Not linkable

☛ Generator Efficiency

Type: Table Slot
Units: FRACTION
Description: the efficiency of the generators in producing power
Information: This value is the fraction of the maximum theoretical power which could be obtained from an ideal turbine.
I/O: Required input
Links: Not linkable

☛ Maximum Turbine Power

Type: Table Slot

Units: POWER
Description: maximum turbine power output
Information:
I/O: Required input
Links: Not linkable

☛ Min and Max Operating Head
Type: Table Slot
Units: LENGTH
Description: the minimum and maximum operating head for the turbines
Information:
I/O: Required input
Links: Not linkable

☛ Number of Units
Type: Table Slot
Units: NONE
Description: integer number of turbines in plant
Information:
I/O: Required input
Links: Not linkable

☛ Peak Flow
Type: Series Slot
Units: FLOW
Description: most efficient flow through turbines for the current Operating Head
Information:
I/O: Output only
Links: Not linkable

☛ Peak Hours
Type: Series Slot
Units: TIME
Description: the number of hours operated at peak flow
Information:
I/O: Output only
Links: Not linkable

☛ Plant Efficiency
Type: Series Slot

Units: FRACTION
Description: ratio of actual power produced to peak and base theoretical power
Information:
I/O: Output only
Links: Not linkable

☛ Power Capacity

Type: Agg Series Slot
Units: POWER
Description: power that could be produced if flow is directed through the turbines given the operating head.
Information: Calculated by the two peak power methods and cannot be input by the user.
I/O: Output only
Links: Not linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: FRACTION
Description: the percentage of full capacity of the turbine units in the hydropower plant
Information: The value of this slot defaults to 100% if not input by user.
I/O: Required input
Links: Not linkable

First, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction if necessary (the default is 1.0) and checks for plant shutoff/failure.

If the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Energy, Power, Peak Flow, Peak Hours, Power Capacity, and Plant Efficiency are set equal to zero.
2. Turbine Release is set equal to zero.
3. Tailwater and Operating Head are computed and set. No further computations are performed.

If Energy or Turbine Release are input by the user, an error is posted. These are not valid input slots for the Peak Power method. Peak Flow and Peak Power are then calculated as follows:

1. The Maximum Turbine Power is then used in the Best Generator Power table interpolate a value for headAtMaxPower.

2. headAtMaxPower is used in the Best Generator Flow table to interpolate a value for flowAtMaxPower. Flow is temporarily set as flowAtMaxPower.

3. efficiencyAtMaxPower is calculated by the following formula:

$$\text{efficiencyAtMaxPower} = \text{Maximum Turbine Power} \times \frac{1000000}{\text{flowAtMaxPower} \times \text{headAtMaxPower} \times 999.99 \times 9.79908}$$

where 999.99 is the density of water (Kg/M^3) at five degrees C and 9.79908 is gravitational acceleration (M/s^2) at 37 degrees North latitude.

4. **RiverWare**™ then iterates until all of the following conditions are met; the absolute difference between Qnew and flow is less than 5 cfs, the number of iterations is greater than the maximum number of iterations, and the Operating Head is less than the Minimum Operating Head. Initially, Qnew is set equal to flow. Plant Flow is calculated as Qnew times the Number of Units times the Plant Power Cap Fraction. Then, the Tailwater method selected by the user is performed. If the Operating Head is greater than the Maximum Operating Head, the Operating Head is set equal to the Maximum Operating Head. If the Operating Head is less than the Minimum Operating Head, Operating Head is set equal to Minimum Operating Head. If the Operating Head is greater than headAtMaxPower, flow is calculated as:

$$\text{flow} = \text{Maximum Turbine Power} \times \frac{1000000}{\text{Operating Head} \times \text{efficiencyAtMaxPower} \times 999.99 \times 9.79908}$$

Otherwise, flow is determined by interpolation using Operating Head and the Best Generator Flow table. This set of calculations is repeated until the iteration criteria described above are met.

5. If Operating head is less than the Minimum Operating Head, Turbine Release, Energy, Power, Power Capacity, Peak Flow, Peak Hours, and flow are all set to zero. Then, the Tailwater method is re-executed. If Operating Head is greater than headAtMaxPower, Peak Power is set equal to Maximum Turbine Power. Otherwise, Peak Power is determined from the Best Generator Power table using Operating Head.

Once flow and Peak Power are determined, the following computations are performed:

$$\text{Peak Flow} = \text{flow} \times \text{number of Units} \times \text{Power Plant Cap Fraction}$$

$$\text{Plant Peak Power} = \text{Peak Power} \times \text{Number of Units} \times \text{Power Plant Cap Fraction}$$

$$\text{Peak Flow Volume} = (\text{Outflow} - \text{Spill}) \times \text{Timestep (in seconds)}$$

$$\text{Peak Hours} = \frac{\text{Peak Flow Volume}}{\text{Peak Flow} \times 3600 \text{ seconds}}$$

If the value of Peak Hours is greater than the length of the timestep, Peak Hours is set to the length of the timestep, Turbine Release is equal to Peak Flow, and Total Controlled Release is equal to Outflow minus Unregulated Spill. Otherwise, Peak Hours remains unchanged, Turbine Release equals Outflow minus spill, and Total Controlled Release equals Peak Flow.

Next, the theoretical and actual energy production is calculated.

$$\text{Peak Energy} = \text{Peak Hours} \times \text{Plant Peak Power} \times \text{Generator Efficiengy}$$

$$\text{Best Peak Theoretical} = \text{Peak Flow} \times \text{Operating Head} \times \text{Peak Hours} \times 0.00980229$$

The value, 0.00980229, is a conversion factor necessary for energy to have units of megawatt-hours.

$$\text{Energy (MWH)} = 62.4\ \text{(lb/ft3)} \times Q\ \text{(cfs)} \times \text{time (hrs)} \times \text{Head (ft)} \times 1\ \text{KW} / 737.56\ \text{(ft-lb/s)} \times 1\ \text{MW} / 1000\ \text{KW}$$

For Q in cms and Head in meters the final conversion is 0.00980229.

Finally, the following slots are set:

$$\text{Power Capacity} = \text{Plant Peak Power}$$

$$\text{Energy} = \text{Peak Energy}$$

$$\text{Power} = \frac{\text{Energy}}{\text{timestep length (hours)}}$$

If Best Peak Theoretical is equal to zero, Plant Efficiency is also equal to zero. Otherwise Plant Efficiency is calculated as:

$$\text{Plant Efficiency} = \frac{\text{Energy}}{\text{Best Peak Theoretical}}$$

If a spill method is selected which utilizes the Spilled Energy Power Coefficient and this value is not input by the user, it is set as:

$$\text{Spilled Energy Power Coefficient} = \frac{\text{Power}}{\text{Turbine Release}}$$

If Turbine Release is zero, the Spilled Energy Power Coefficient is also zero.

21.1.1.9 Peak Power Equation

The Peak Power Equation method provides a standard equation method of calculating plant peaking power for a portion of the computational timestep using the water power equation.

Slots Specific to this Method

☛ Head Loss

Type: Table Slot
Units: LENGTH
Description: The head loss water incurs before it reaches the turbines.
Information: The slot is set to zero if not input by the user.
I/O: optional
Links: Not linkable

☛ Min Power Elevation

Type: Table Slot
Units: LENGTH
Description: Minimum pool elevation at which power can be generated
Information: Single value in a 1x1 table slot
I/O: Required input
Links: Not linkable

☛ Net Head vs. Peak Release

Type: Table Slot
Units: LENGTH VS. FLOW
Description: relationship between the net head and the maximum possible turbine release
Information: Net Head must account for any head loss
I/O: Required input
Links: Not linkable

☛ Peak Release

Type: Series Slot
Units: FLOW
Description: The flow through the turbines when the plant is operating at peak capacity
Information: Peak Release is solved for iteratively using net head, tailwater elevation and pool elevation
I/O: Output only
Links: Not linkable

☛ Peak Times

Type: Series Slot
Units: TIME
Description: The time at which the plant is operating at peak capacity
Information: Peak Time is calculated as the timestep flow volume divided by the Peak Release.
I/O: Output
Links: Not linkable

☛ Plant Efficiency

Type: Table Slot
Units: NONE
Description: the decimal percent efficiency at which the plant is operating
Information: Single value in a 1x1 table slot. Plant efficiency should incorporate both generator efficiency and turbine efficiency.
I/O: Defaults to 1.0 if not input. Must be between 0 and 1.
Links: Not linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: NONE
Description: decimal fraction of the power capacity at which the plant is operating
Information: Used in the case of outages or reductions in the plant operating capacity.
I/O: Defaults to 1.0 if not user input. Must be between 0 and 1.
Links: Not linkable

The Peak Power Equation method first performs a series of checks.

The selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the Minimum Power Elevation was input by the user. If no value was input, a **RiverWare**™ error is posted and the simulation run is aborted. If the previous Pool Elevation is less than the Minimum Power Elevation or the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Power, Energy, Peak Release, and Peak Time are set equal to zero.

2. If the Turbine Release is input or already set from the Dispatch Method “solveMB_givenInflowRelease,” a **RiverWare**™ error is flagged and the run is aborted.

3. Turbine Release is set equal to zero.
4. Tailwater and operating head are computed. No further calculations are performed.

The method checks whether Turbine Release (the average flow through the turbines over the whole timestep) is user input or set by rules. (i.e. the method checks if the dispatch type is solveMB_givenInflowRelease.) If the given Turbine Release value is greater than Outflow minus Spill, an error is posted and the run is aborted.

If Turbine Release is not user input or set by rules, it is calculated as the minimum of MaxTurbine Release and Outflow minus Spill,

$$TurbineRelease = Min(MaxTurbineRelease, Outflow - Spill)$$

where the Max Turbine Release is interpolated from the Peak Release vs. Net Head Table, given the Net Head over the entire timestep.

In order to calculate the time at peak production, the flow which passes through the turbines during this time period must be calculated. This Peak Release is the maximum possible flow through the turbines given the Net Head and will be solved for iteratively as described in the steps below.

1. Peak Release is initially set to zero.
2. Tailwater Elevation is determined using Peak Release + Spill as the “flow” value in the selected Tailwater method. (If the Turbine Release slot is linked, it can be assumed that Spill is sent elsewhere and does not affect Tailwater so the “flow” value should be set to Peak Release only.)
3. The operating head is calculated as the Pool Elevation minus the Tailwater Elevation.
4. The Net Head is calculated as the operating head minus the head loss.
5. Given the Net Head, the Peak Release is interpolated from the Net Head vs. Peak Release table and then multiplied by the Power Plant Cap Fraction.
6. The new Peak Release value is compared with the previous value and the iteration will continue until the value converges. (Note: Convergence Percentage is a general slot on power reservoirs representing the convergence in all iterative solutions-- the slot defaults to 0.0001 if not input.)

Once Peak Release is calculated, the Peak Time will be solved for as the volume of flow that passes through the turbines in a timestep divided by the Peak Release:

$$\text{Peak Time} = \frac{\text{Timestep Flow Volume}}{\text{Peak Relase}}$$

where timestepFlowVolume is an internal variable defined as:

$$\text{Timestep Flow Volume} = \text{Turbine Release} \times \text{timestep seconds}$$

RiverWare checks that the Peak Time is not greater than the timestep length. If it is, the run aborts and an error is posted.

Power is calculated with the standard water power equation. The Peak Power Equation method uses Peak Release as the flow value and Net Head at Peak Release as the head value.

$$Power = \frac{\text{Peak Release} \times \text{Net Head} \times \text{efficiency} \times \text{Plant Cap Fraction}}{\text{Unit Compatibility Factor}}$$

The unit compatibility factor comes from balancing units and is 102.01697767 in internal RiverWare units.

Energy is finally computed as the product of Power and Peak Time:

$$Energy = Power \times \text{Peak Time}$$

21.1.1.10 Peak Power Equation with Off Peak Spill

The Peak Power Equation with Off Peak Spill method provides a standard equation to calculate peaking power for a portion of the computational timestep using the water power equation. Included also is a calculation of the off peak spill that occurs when the turbines are not operating.

Slots Specific to this Method

☛ Head Loss

Type: Table Slot
Units: LENGTH
Description: The head loss water incurs before it reaches the turbines.
Information: The slot is set to zero if not input by the user.
I/O: optional
Links: Not linkable

☛ Maximum Power Pool Drawdown

Type: Scalar
Units: LENGTH
Description: maximum vertical drop permitted in the power pool in one timestep for power release
Information:

I/O: Required input

Links: Not Linkable

☛ Minimum Power Elevation

Type: Table Slot
Units: LENGTH
Description: Minimum pool elevation at which power can be generated.
Information: Single value in a 1x1 table slot
I/O: Required input
Links: Not linkable

☛ Minimum Elevation for Power Operations

Type: Series Slot with Periodic Input, HERE (Slots.pdf, Section 4.8)
Units: LENGTH
Description: Minimum pool elevation at which power operations can occur.
Information: This slot provides another way to limit the additional proposed hydropower release described HERE (USACE_SWD.pdf, Section 3.9). In simulation, a warning is issued if the Pool Elevation is below this elevation.
I/O: Optional input as either series or periodic values.
Links: Not linkable

☛ Net Head vs Plant Efficiency

Type: Table Slot
Units: LENGTH VS NONE
Description: relationship between the Net Head and the efficiency of the plant
Information: Net Head includes head loss and Efficiency includes both generator and turbine efficiency
I/O: Required Input
Links: Not Linkable

☛ Net Head vs. Generator Capacity

Type: Table Slot
Units: LENGTH VS. POWER
Description: relationship between Net Head and the maximum possible power produced
Information: Net Head includes head loss
I/O: Required Input
Links: Not Linkable

☛ Off Peak Spill

Type: Series Slot

Units: FLOW
Description: The spill that occurs during the off peak portion of the timestep.
Information: Off Peak Spill is the fraction of the spill that occurs when power is not being produced. The time weighted average of Off Peak Spill and Peak Spill equals the Spill. If the Peak Time equals the timestep length, Off Peak Spill is NaN.
I/O: Output only
Links: Not linkable

PEAK RELEASE

Type: Series Slot
Units: FLOW
Description: The flow through the turbines when the plant is operating at generator capacity
Information: Peak Release is solved for iteratively using net head, tailwater elevation and pool elevation
I/O: Output only
Links: Not linkable

PEAK TIME

Type: Series Slot
Units: TIME
Description: The time at which the plant is operating at peak capacity
Information: Peak Time is calculated as the timestep flow volume divided by the Peak Release.
I/O: Output
Links: Not linkable

PLANT POWER LIMIT

Type: Series Slot
Units: POWER
Description: The user specified upper limit on power production.
Information: If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit and the Energy is recalculated. A new Turbine Release is then calculated based on the Plant Power Limit.
I/O: Optional Input
Links: Not Linkable

PEAK SPILL

Type: Series Slot
Units: FLOW
Description: The spill that occurs during the Peak Time.

Information: Peak Spill is the portion of the Spill that occurs during the Peak Time. The time weighted average of Peak Spill and Off Peak Spill equals the Spill.
I/O: Output
Links: Not linkable

☛ POWER PLANT CAP FRACTION

Type: Series Slot
Units: NONE
Description: decimal fraction of the power capacity at which the plant is operating
Information: Used in the case of outages or reductions in the plant operating capacity.
I/O: Defaults to 1.0 if not user input. Must be between 0 and 1.
Links: Not linkable

This method is called from the dispatch method, typically after Outflow, Storage, and Pool Elevation have been calculated. If Energy is input, then this method is also called from an iterative loop used to determine the Turbine Release, Peak Release, and Peak Time and/or Spill that satisfies the Energy.

The Peak Power Equation with Off Peak Spill method performs a series of checks. First, the method checks if maximum drawdown is exceeded for two cases. In the first case, if there is a valid Top of Conservation Pool slot, i.e. this is a U.S. Army Corp of Engineers model, the method checks if the Pool Elevation is greater than the Top of Conservation Pool. If so, no error or warning is posted; the drawdown limitation only applies to the conservation pool. If the drawdown is exceeded and the Pool Elevation is less than the top of conservation pool, an error is posted and the run is aborted. Note, an abortive error is only posted if the reservoir is dispatching at the current controller timestep. If it is dispatching at a forecast timestep and max power pool drawdown is exceeded, no error is issued. It is assumed that either the inflow or outflow will be modified when the reservoir dispatches at the current timestep and would catch any errors then.

In the second case, if there is no Top of Conservation Pool slot (i.e. a non U.S. Army Corp of Engineers model), if the calculated Pool Elevation results in exceeding the Maximum Power Pool Drawdown, a warning is posted.

If the calculated Pool Elevation is less than the value in the Minimum Elevation for Power Operations, a warning message is posted.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the Minimum Power Elevation was input by the user. If no value was input, a **RiverWare**™ error is posted and the simulation run is aborted. If the previous Pool Elevation is less than the Minimum Power Elevation or the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Power, Energy, Peak Release, and Peak Time are set equal to zero.
2. If the Turbine Release is input or already set from the Dispatch Method "solveMB_givenInflowRelease," a **RiverWare**™ error is flagged and the run is aborted.
3. Turbine Release is set equal to zero.
4. Tailwater and operating head are computed. No further calculations are performed.

The method checks whether Turbine Release (the average flow through the turbines over the whole timestep) is user input or set by rules. (i.e. the method checks if the dispatch type is solveMB_givenInflowRelease.) If so, and if the given Turbine Release value is greater than Outflow minus Spill, an error is posted and the run is aborted.

In order to calculate the time at peak production, the flow which passes through the turbines during this time period must be calculated. This Peak Release is the maximum possible flow through the turbines given the Net Head and will be solved for iteratively as described in the steps below.

1. Peak Release is initially set to zero.
2. Given the net head from the previous timestep (Operating Head at previous timestep minus Head Loss), the efficiency is interpolated from the Net Head vs Efficiency table. The previous Operating Head is used as an approximation so as not to introduce an additional variable in the iteration. As a result, the Tailwater Elevation at the initial timestep must be input. The net head for the initial timestep is the initial Pool Elevation minus the initial Tailwater Elevation minus Head Loss.
3. The current Tailwater Elevation is determined using the maximum of Peak Release plus Unregulated Spill or the current Outflow as the flow value in the selected Tailwater method. (Note that if Energy is input, the Tailwater Elevation slot value shown will be calculated using the average Outflow from the timestep, but the tailwater elevation used in the Peak Release calculation is calculated as described here. If Energy is not input, the Tailwater Elevation slot value will be the peak Tailwater Elevation calculated here.)
4. The Operating Head is calculated as the average Pool Elevation minus the Tailwater Elevation.
5. The Net Head is calculated as the Operating Head minus the Head Loss.
6. Given the Net Head, the Generator Capacity is interpolated from the Net Head vs. Generator Capacity table and is then multiplied by the Power Plant Cap Fraction. If this new capacity is greater than the Plant Power Limit, if valid, the generator capacity is reset to the Plant Power Limit.

7. Peak Release is calculated according to the power equation. The unit compatibility factor comes from balancing units and the specific weight of water; it is 102.01697767 in internal RiverWare units.

$$\text{Peak Release} = \frac{\text{Generator Capacity} \times \text{Unit Compatibility Factor}}{\text{Net Head} \times \text{Efficiency}}$$

The new Peak Release value is compared with the previous value and the iteration, steps 3-7, continue until the value converges. (Note: Convergence Percentage is a general slot on power reservoirs representing the convergence in all iterative solutions-- the slot defaults to 0.0001 if not input.)

If Turbine Release is not user input or set by rules (not in the dispatch method solveMB_givenInflowRelease), TempTurbineRelease is calculated as the minimum of the Peak Release and Outflow minus Spill,

$$TempTurbineRelease = Min(PeakRelease, Outflow - Spill)$$

The Spill will be non-zero only if there is Unregulated Spill or a spill value is set by user input or rules. Once Peak Release is calculated, the Peak Time will be solved for as the volume of flow that passes through the turbines in a timestep divided by the Peak Release:

$$\text{Peak Time} = \frac{\text{TempTurbine Release} \times \text{timestep seconds}}{\text{Peak Release}}$$

RiverWare checks that the Peak Time is not greater than the timestep length. If it is, the run aborts and an error is posted. Next power is set to be the Generator Capacity:

$$Power = GeneratorCapacity$$

Turbine Release is the Peak Release averaged over the whole timestep:

$$TurbineRelease = \frac{PeakRelease \times PeakTime}{TimestepSeconds}$$

Energy is computed as the product of Power and Peak Time:

$$Energy = Power \times \text{Peak Time}$$

Peak Spill and Off Peak Spill are then determined based on Peak Time, Spill, and Unregulated Spill. If Unregulated Spill is non-zero, then Peak Spill is assumed to be equal to Unregulated Spill. Unregulated spill is calculated based on the pool elevation and occurs over the entire timestep. Off Peak Spill is the sum of Unregulated Spill and the Regulated Spill plus Bypass apportioned over the off peak time. If there is no Unregulated Spill, the

Peak Spill is zero, and the Off Peak Spill is the Regulated Spill plus Bypass apportioned over the off peak time. If the Peak Time is equal to the timestep length, then Peak Spill is equal to Spill and Off Peak Spill remains NaN.

Finally, if the Load slot is visible and valid, the Thermal Purchase, Dump Energy and Operation Factor are calculated. See the Load Calculation Section on page 624 for more information on these slots.

21.1.1.11 LCR Power

The LCR Power method uses an empirical relationship to calculate the energy produced by the Hoover, Davis, and Parker dams on the Lower Colorado River. The method replicates the calculations from the U.S. Bureau of Reclamation BHOPS FORTRAN program. Energy is calculated as a function of flow, Operating Head, Plant Efficiency, and the Power Coefficients.

Slots Specific to this Method

☛ LCR Input Efficiency

Type: Series Slot
Units: NONE
Description: a fractional value ranging from 0 to 1 which may be used to scale the efficiency or the turbine units in the hydropower plant.
Information:
I/O: Optional; the value defaults to 1 if not input by the user.
Links: Not linkable

☛ Lower Colo Power Coeffs

Type: Table Slot
Units: NONE
Description: two values used as empirical coefficients in relating flow, head, and efficiency to energy.
Information: The coefficients for Hoover Dam are empirically derived. For Davis and Parker dams, they reduce to coeff1=1 and coeff2=0.
I/O: Required input
Links: Not linkable

☛ Net Energy Request

Type: Series Slot
Units: ENERGY
Description: represents the total energy requested by the grid
Information: This slot is only used when a Best Efficiency flag is set in the Energy slot.
I/O: Optional; only used when a Best Efficiency flag is set in the Energy slot.

Links: Not linkable

☛ Plant Efficiency

Type: Series Slot
Units: NONE
Description: a fractional value ranging from 0 to 1 that represents the percentage of full efficiency of the turbine units in the hydropower plant
Information: In the case of Davis and Parker dams, this equals the LCR Input Efficiency. For Hoover Dam, a Plant Efficiency is calculated.
I/O: Output only
Links: Not linkable

☛ Power Coefficient

Type: Series Slot
Units: POWER PER FLOW
Description: power generated per unit power release
Information: This coefficient corresponds to the efficiency of the entire plant. It is not used in calculation and is displayed only for the benefit of the user.
I/O: Output only
Links: Not linkable

☛ Station Energy Table

Type: Table Slot
Units: ENERGY
Description: represents the energy required to run the station for each day of the week
Information: This slot is only used when a Best Efficiency flag is set in the Energy slot.
I/O: Optional; only used when a Best Efficiency flag is set in the Energy slot.
Links: Not linkable

The first step in the LCR Power method is to determine the Operating Head. This is accomplished by executing the Tailwater method specified by the user.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Power, Energy, Plant Efficiency, and Power Coefficient are set equal to zero.

2. If the Turbine Release is input or already set from the Dispatch Method “solveMB_givenInflowRelease,” a **RiverWare**™ error is flagged and the run is aborted.

3. Turbine Release is set equal to zero. No further calculations are performed.

If Energy is input by the user or it has been flagged as Best Efficiency, the LCR Power Release method is called to calculate the Turbine Release. This method takes a power request, determines if it can be met given the maximum power that can be generated for the given head, and sets the Turbine Release required to generate the requested power. If the Turbine Release is not already calculated by the LCR Power Release method, it is set as Outflow minus Spill. The value of Turbine Release is checked against the maximum value set by the user (if it has been set). If Turbine Release is greater than the maximum value an error is posted which reads, “Turbine Release required to meet Energy request is greater than the maximum Turbine Release.”

Power and Energy are then calculated by the following equations:

$$\text{Energy (MWH)} = \left(\text{Lower Colo Power Coeffs\_1} \times \frac{62.4}{737.5} \times \text{Outflow (1000 cfs)} \times \text{Timestep (hours)} \times \frac{\text{Operating Head (ft)}}{1000} - \text{Lower Colo Power Coeffs\_2}\right) \times \text{LCR Input Efficiency} \times 1000$$

where 62.4 is the unit weight of water in pounds per cubic foot and 737.5 represents ft.-lb./sec. per Kilowatt.

$$\text{Power} = \frac{\text{Energy}}{\text{Timestep (hours)}}$$

If energy is zero, Plant Efficiency and the Power Coefficient are also zero. Otherwise they are calculated as:

$$\text{Plant Efficiency} = \frac{\text{Energy}/1000}{\frac{62.4}{737.5} \times \text{Outflow (1000 cfs)} \times \text{Timestep (hours)} \times \frac{\text{Operating Head (ft)}}{1000}}$$

$$\text{Power Coefficient} = \frac{\text{Energy}}{\text{Turbine Release} \times \text{Timestep (hours)}}$$

21.1.1.12 Unit Power Table

This method uses a 3-D table that contains the columns Operating Head, Turbine Release, and Power for **each unit** in the plant.

SLOTS ADDED BY THIS METHOD

Note, many of these slots have column or row dimensions based on the number of units. The rows/columns of these slots are expanded at the beginning of the run to match the value in the Number of Units slot. When first configuring this method, the user must enter the

Number of Units, then run the model (stepping through 1 timestep is enough) to grow the slots to the right dimensions.

The following slots are instantiated when this method is selected.

Auto Unit Best Turbine Q Table

Type: Table Slot
Units: Length, flow
Description: Table showing most efficient release levels for given operating heads
Information: This table is generated from the Unit Power Table. The first column for each block is Operating Head and second column is Turbine Release. It will have one block for each unit.
I/O: Automatically generated at beginning of run
Links: Not Linkable

Auto Unit Max Turbine Q Table

Type: Table Slot
Units: Length, Flow
Description: Table showing maximum release possible for given operating heads
Information: This table is generated from the Unit Power Table. The first column for each block is Operating Head and the second is Turbine Release. It will have one block for each unit.
I/O: Automatically generated
Links: Not Linkable

Minimum Power Elevation

Type: Table Slot
Units: Length
Description: The minimum elevation at which the reservoir can still produce power.
Information:
I/O: Optional Input Only
Links: Not Linkable

Number of Units

Type: Table Slot
Units: None
Description: Number of units in the plant
Information: This key scalar slot (existing slot, 1x1 table) indicates the number of units (turbines) at a power reservoir. The dimensions of several other slots are directly related to this value; specifically, both input data represented in Table Slots and unit-level series data represented in Agg Series Slots are require

one row or column for each unit. At the **beginning of each run**, RiverWare will confirm that the value for the Number of Units slots is consistent with the dimensions of related slots. If any inconsistencies are detected, the relevant slots are resized as appropriate. If additional input data are required, the user is notified and the run is aborted.

I/O: Required Input only

Links: NA

Number of Units Generating

Type: Series Slot

Units: None

Description: Number of units that are generating at a given timestep

Information: The value is the sum of the Unit Is Generating

I/O: Output only

Links: NA

Power Curvature Tolerance

Type: Scalar Slot

Units: None

Description: The power curvature tolerance is used to account for anomalies in Unit Power Table data and round off error while calculating slopes.

Information: Although the units for the slot are “None”, the comparison is implicitly using (MW/cms)

I/O: Input or defaults to 1X10-6

Links: Links: Not linkable

Unit Is Generating

Type: Agg Series Slot

Units: None

Description: This slot is used to control whether units are available.

Information: There is one column for each unit. Before a run, an input value of 1 indicates that the unit **must** generate power at that date; otherwise an input value of 0 indicates that the unit can **not** generate power at that timestep. A NaN indicates that the unit is available and the model will decide if it can generate. At the end of a run, an output 1 indicates the unit generated power, a 0 indicates it did not.

I/O: Can be input by user

Links: Not Linkable

Unit Energy

Type: Agg Series Slot

Units: Energy
Description: Energy produced by each unit
Information: There is one column for each unit. A value indicates the energy being generated by the unit at that timestep, and takes into account frequency regulation. A negative value can be input or set by a rule to represent a unit that is spinning, motoring, or condensing (actually consuming energy).
I/O: Input, Rules, or Output
Links: Not linkable

Unit Power

Type: Agg Series Slot
Units: Power
Description: The power that is generated by each unit
Information: There will be one column for each unit. A value indicates the power being generated by the unit at that timestep, and takes into account losses due to frequency regulation.
I/O: Calculated
Links: Not linkable

Unit Power Table

Type: Table Slot
Units: Length, Flow, Power, Flow, Power, etc...
Description: A 3 dimensional table relating operating head, turbine release, and power for each unit in the plant. There will be 1 block (3 columns) for each unit.
Information: The last row for each operating head represents the **max capacity**. Best efficiency is automatically calculated. The three values in a given row and unit block represent a legal operating point for that unit, i.e., the Power which that unit would generate at that head and turbine flow. It will be necessary to enforce that a point of zero flow and zero power production be entered in the table for each operating head. It is also required that this table be concave.
I/O: Required Input only
Links: NA

Unit 1			Unit 2		
Operating Head (ft)	Turbine Flow (1000 cfs)	Power (KW)	Operating Head (ft)	Turbine Flow (1000 cfs)	Power (KW)
100	0	0	99	0	0
100	10	2000	99	10	1000
100	20	3000	99	20	2000

Unit 1			Unit 2		
Operating Head (ft)	Turbine Flow (1000 cfs)	Power (KW)	Operating Head (ft)	Turbine Flow (1000 cfs)	Power (KW)
100	30	4000	99	30	3000
200	0	0	200	0	0
200	10	2500	200	10	1700
200	20	3500	200	20	2500
200	25	3800	200	25	2800
200	30	4500	200	30	3500
300	0	0	295	0	0
300	10	3000	295	10	3000
300	25	5000	295	25	4000

☛ Unit Priority Table

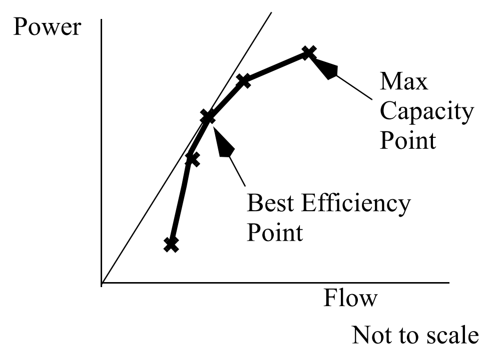
Type: Table Slot
Units: No Units
Description: The priority that each unit is started or stopped in the power plant
Information: There will be one row for each unit. In optimization only, units with lower numerical values are higher priority and are scheduled to release power in preference to lower priority units. If a value is absent then that unit is given the lowest priority. For units with equal priority, the unit efficiency will determine precedence, i.e. a unit with a higher efficiency will be prioritized higher than other units with the same priority value (note, not implemented yet. Currently, units with equal priority are turned on/off in an arbitrary order.). Currently, this table is only used in optimization. It is not used in simulation.
I/O: Optional input
Links: NA

☛ Unit Turbine Release

Type: Agg Series Slot
Units: Flow
Description: Flow through each unit
Information: There is one column for each unit. The value is the expected Turbine Release through the unit at that timestep.
I/O: Input, Rules, Output
Links: Not Linkable

Method Details

This method will use table interpolation to calculate Power and Energy at a known Operating Head based on the characteristics of each unit. This method, Unit Power Table, works in a similar manner to the current method, Plant Efficiency Curve.



At the beginning of the run, the method creates the Auto Unit Best and Auto Unit Max Turbine Q tables from the Unit Power Table. The Auto Unit Max Turbine Q table are the points from the Unit Power Table that correspond to the largest Turbine Release for a given Operating Head. The Auto Unit Best Turbine Q table is calculated from the Unit Power Table as follows: For each unit and each operating head, the method determines the point (flow, power) that corresponds to a line drawn from the origin and it tangent to the curve. This tangent point is determined by calculating the slope of the line from each point to the origin; the point with the largest slope is the tangent. Then the operating head and turbine flow for this point are added to the Auto Unit Best Turbine Q.

The description of the solution in this section assumes that mass balance has already occurred (i.e. Inflow, Outflow, Storage, Pool Elevation have been calculated) and the method is trying to compute the Energy and Power produced by that Outflow. At the start of this method, there is also an estimate of the Turbine Release calculated as

$$\text{Turbine Release} = \text{Outflow} - (\text{Unregulated Spill} + \text{Regulated Spill} + \text{Bypass})$$

where the spills are either known or estimated based on the current Pool Elevation. This Turbine Release may be reset if it cannot be met. If Energy is input, set by a rule, or flagged **BEST EFFICIENCY** (B) or **MAX CAPACITY** (M) or **UNIT** (U) then Energy and Power are solved using the method **Unit Power Table Release** described in Section 21.1.2.8 on page 759.

In this description, “t” indicates the current timestep and “u” indicates that the method will do this for each unit.

The method calculates the tailwater and operating head using the selected method and current release and pool elevation.

For each unit, the previous pool elevation will be compared to the unit’s minimum power elevation.

if (Pool Elevation[t-1] < Unit Minimum Power Elevation[u])
{
 {
 Either set the following to zero or make sure that they are zero:

- Unit Turbine Release[t,u],

- Unit Energy[t,u],
- Unit Power[t,u], and
- Unit is Generating[t,u].

If any of these are non-zero, an error is issued.
}
}

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Power, Energy, and Turbine Release are set equal to zero.
2. Either set the following to zero or make sure that they are zero:
   - Unit Turbine Release[t,u],
   - Unit Energy[t,u],
   - Unit Power[t,u], and
   - Unit is Generating[t,u].

   If any of these are non-zero, an error is issued No further computations are performed.

Next for each unit, an estimate of max flow through all the turbines is calculated as follows: estimate a temporary variable maxPowerRelease (flow) using the Auto Unit Max Turbine Q table. This table contains the columns Operating Head and Turbine Capacity. Because Operating Head is known at the current timestep, table interpolation is used to calculate maxPowerRelease for the given average Operating Head.

If Outflow is set to **Max Capacity** flag, set the Unit Turbine Release to the maximum calculated and compute the power produced by those flows.

Otherwise, if Unit Turbine Release is input/rules for any of the units:

If Unit is Generating is set (input/rules) to 0 for a unit that has a Unit Turbine release, issue an error.

If Unit is Generating is set (input/rules) to 1 for a unit that does not have a Unit Turbine release, issue an error.

If Turbine Release is not set by the U flag, check if Turbine Release = Σ Unit Turbine Release[u]. If they do not match and Turbine Release is input/rules, issue an error. Otherwise, if they do not match, the method resets Turbine Release equal to Σ Unit Turbine Release[u].

If Turbine Release does have the U flag, the Turbine Release is set equal to Σ Unit Turbine Release[u].

If Turbine Release is now greater than Σ maxPowerRelease[u], an error is issued as the specified unit turbine releases cannot be met.

If a regulation method is selected, call it here, otherwise, given the known Unit Turbine Releases[u], the method then looks up the unit flow and operating head on the Unit Power Table to determine the power produced by each unit: Unit Power[u].

Finally, Unit Energy[u] = Unit Power[u] * Time (hrs)

The total Power = Σ Unit Power[u] and total Energy = Σ Unit Energy [u]

Else Unit Turbine Release is not input/rules

Exit the method as there is no way to compute energy/power at a unit level. The dispatch method will continue but no power related slots (e.g. Energy, Unit Energy, Power, Unit Power, Unit Turbine Release, Unit Is Generating, Unit Startup, Unit Shutdown, etc) will be set. Turbine Release is set to the minimum of (Outflow - Spill or MaxPowerRelease). This can happen when dispatching given Inflow and Pool Elevation or Storage. and no Turbine Release or Unit Turbine Release is specified.

Finally, the method computes the slot Unit is Generating based on the Unit Turbine Release and Unit Energy. For each unit, if these are non zero, the Unit is Generating is set to 1. If they are zero, Unit is Generating is set to 0. No inputs are overwritten. Then, the Number of Units Generating is computed as the sum over the columns of the Unit is Generating slot.

21.1.2 Power Release

When Energy is specified, the Power Release method is used to calculate the Turbine Release required. If the Energy request can not be met, the user is notified. There is one method per Power method.

The Power Release category is available when any of the Power methods is selected **except** None, Peak Power, Peak and Base, or Peak Power Equation.

21.1.2.1 None

This is the default method in the Power Release category. No calculations are performed in this method. There are no slots specifically associated with this method. If this method is selected for the Power Release category, a **RiverWare**™ error will be posted and the simulation run will be aborted. A viable power release method must be selected when the Power Release category is visible.

SLOTS SPECIFIC TO THIS METHOD

☛ **NONE**

21.1.2.2 Plant Power Coefficient Release

The Plant Power Coefficient Release method calculates Turbine Release using the entire plant characteristics when Energy is specified. The Plant Power Coefficient Release method is only available if the Plant Power Coefficient method is selected in the Power category. Energy must be input for this method to execute. If Energy is flagged as either **MAX CAPACITY** or **BEST EFFICIENCY**, it is considered input. If Energy is flagged as **MAX CAPACITY**, Turbine Release is set to meet the Energy request at the maximum flow rate. If Energy is flagged as **BEST EFFICIENCY**, Turbine Release is set to meet the Energy request at the most efficient flow rate. If Energy is neither flagged as **MAX CAPACITY** nor flagged as **BEST EFFICIENCY**, the Turbine Release is calculated from the Energy request and a Power Coefficient. The Power Coefficient may be input by the user or calculated by **RiverWare**™ from interpolation of the Best and Max Power Coefficient tables.

If Energy is flagged **UNIT VALUES** (U), and error is issued. This flag is only available with the “Unit Power Table Release” method.

SLOTS SPECIFIC TO THIS METHOD

☛ **NONE**

The first step in the Plant Power Coefficient Release algorithm is to set the Power Plant Cap Fraction to 1.0 if it is not already known.

If the Energy slot is flagged as MAX CAPACITY, the following steps are taken:

1. Qtemp, a local variable, is calculated from interpolation of the Max Turbine Q table using the Operating Head.
2. Turbine Release is calculated with the following equation:

$$\text{Turbine Release} = \text{Qtemp} \times \text{Power Plant Cap Fraction}$$

3. PCmax, a local variable, is determined from interpolation of the Max Power Coefficient table using the Operating Head. The Power Coefficient is set as PCmax if it is not input.
4. Power and Energy are then calculated using the following equations:

$$\text{Power} = \text{Turbine Release} \times \text{PCmax}$$

$$\text{Energy} = \text{Power} \times \text{Timestep (in hours)}$$

5. If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit and the Energy is recalculated. A new Power Coefficient and Turbine Release are then calculated based on the Plant Power Limit.

If Energy is flagged as BEST EFFICIENCY, the following steps are taken:

1. Qtemp is calculated from interpolation of the Best Turbine Q table using the Operating Head.
2. Turbine Release is computed using the following equation:

$$\text{Turbine Release} = \text{Qtemp} \times \text{Power Plant Cap Fraction}$$

3. PCbest, a local variable, is determined from interpolation of the Best Power Coefficient table using the Operating Head. The Power Coefficient is set as PC best if it is not input.
4. Power and Energy are then calculated using the following equations:

$$\text{Power} = \text{Turbine Release} \times \text{PCbest}$$

$$\text{Energy} = \text{Power} - \text{Timestep (in hours)}$$

5. If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit, the Energy is recalculated, and the Turbine Release is recalculated as Plant Power Limit / PCbest.

If Energy is not flagged as either MAX CAPACITY or BEST EFFICIENCY and the Power Coefficient is input, the following steps are taken.

1. If the Power Coefficient is less than 0.00000001, a **RiverWare**™ error is posted and the simulation run is aborted.
2. Power is calculated using the following equation:

$$\text{Power} = \text{Energy}/\text{Timestep}$$

where the Timestep is in hours.

3. Qout, a local variable, is calculated with the following equation:

$$\text{Qout} = \text{Power}/\text{Power Coefficient}$$

4. Qtemp, a local variable, is determined by the interpolation of the Max Turbine Q table using the Operating Head.
5. Qmax, a local variable, is computed using the following equation:

$$\text{Qmax} = \text{Qtemp} \times \text{Power Plant Cap Fraction}$$

6. If Qout is greater than Qmax, the largest discharge value in the Max Turbine Q table is found. If this value is greater than or equal to Qout, Turbine Release is set equal to Qmax. If the value is less than Qout, a **RiverWare**™ error is posted and the simulation run is aborted.
7. If Qout is less than or equal to Qmax, Turbine Release is set equal to Qout.

If Energy is not flagged as either MAX CAPACITY or BEST EFFICIENCY and the Power Coefficient is not given, the following steps are taken:

1. Power is calculated using the following equation:

$$\text{Power} = \text{Energy}/\text{Timestep}$$

where the timestep is in hours.

2. The best and max power coefficients are interpolated using the Operating Head and the Best Power Coefficient and the Max Power Coefficient tables, respectively.
3. QbestTemp and QmaxTemp (local variables) are then determined using the Operating Head to interpolate values from the Best Turbine Q and Max Turbine Q tables, respectively.

4. Qbest, a local variable, is computed using the following equation:

$$\text{Qbest} = \text{QbestTemp} \times \text{Power Plant Cap Fraction}$$

5. Qmax, a local variable, is calculated using the following equation:

$$\text{Qmax} = \text{QmaxTemp} \times \text{Power Plant Cap Fraction}$$

6. If Power divided by the best power coefficient is less than or equal to Qbest, Turbine Release is set equal to Power divided by the best power coefficient.

7. If Power divided by the max power coefficient is greater than Qmax, Turbine Release is set equal to the max turbine flow.

8. If neither 3) nor 4) is true, an interpolated value (pcoeffINTERP) is found between the best and max power coefficients based on how close Power is to both the product of Qbest and the best power coefficient, and the product of Qmax and the max power coefficient. The following pair of equations is used to quantitatively determine the pcoeffINTERP value:

$$\text{pcoeffFRACTION} = \frac{(\text{Power} - \text{best power coefficient} \times \text{Qbest})}{(\text{max power coefficient} \times \text{Qmax} - \text{best power coefficient} \times \text{Qbest})}$$

$$\text{pcoeffINTERP} = \text{best power coefficient} + (\text{max power coefficient} - \text{best power coefficient}) \times \text{pcoeffFRACTION}$$

9. The Turbine Release is then calculated with the following equation:

$$\text{Turbine Release} = \text{Power} / \text{pcoeffINTERP}$$

21.1.2.3 Plant Efficiency Curve Release

The Plant Efficiency Curve Release method calculates Turbine Release using the entire plant characteristics when Energy is specified. The Plant Efficiency Curve Release method is only available if the Plant Efficiency Curve method is selected in the Power category. Energy must be input or set by a rule for this method to execute. If Energy is flagged as either **MAX CAPACITY** or **BEST EFFICIENCY**, it is considered input. If Energy is flagged as **MAX CAPACITY**, Turbine Release is set to meet the Energy request at the maximum flow rate. If Energy is flagged as **BEST EFFICIENCY**, Turbine Release is set to meet the Energy request at the most efficient flow rate. If Energy is neither flagged as **MAX CAPACITY** nor flagged as **BEST EFFICIENCY**, the Turbine Release is calculated from the Energy request.

If Energy is flagged **UNIT VALUES** (U), and error is issued. This flag is only available with the “Unit Power Table Release” method.

Slots Specific to this Method

☛ NONE

The first step in the Plant Efficiency Curve Release algorithm is to set the Power Plant Cap Fraction to 1.0 if it is not already known.

If the Energy slot is flagged as MAX CAPACITY, the following steps are taken:

1. Qtemp, a local variable, is calculated as the maximum release using the Operating Head and the Plant Power Table.
2. Turbine Release is calculated with the following equation:

$$\text{Turbine Release} = \text{Qtemp} \times \text{Power Plant Cap Fraction}$$

3. Power is determined directly from the Plant Power Curve.
4. Energy is calculated as:

$$\text{Energy} = \text{Power} \times \text{Timestep}$$

5. The Power Coefficient is calculated as:

$$PowerCoefficient = (Power)/(TurbineRelease)$$

6. If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit and the Energy is recalculated. A new Power Coefficient and Turbine Release are then calculated based on the Plant Power Limit.

If Energy is flagged as BEST EFFICIENCY, the following steps are taken:

1. Qtemp is computed as the most efficient release given the Operating Head and the Plant Power Table.
2. Turbine Release is computed using the following equation:

$$\text{Turbine Release} = \text{Qtemp} \times \text{Power Plant Cap Fraction}$$

3. Power is determined directly from the Plant Power Curve.
4. Energy is calculated as:

$$\text{Energy} = \text{Power} \times \text{Timestep}$$

5. The Power Coefficient is calculated as:

$$PowerCoefficient = (Power)/(TurbineRelease)$$

6. If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit and the Energy is recalculated. A new Power Coefficient and Turbine Release are then calculated based on the Plant Power Limit.

If Energy is not flagged as either MAX CAPACITY or BEST EFFICIENCY and the Power Coefficient is input, the following steps are taken.

1. If the Power Coefficient is less than 0.00000001, a **RiverWare**™ error is posted and the simulation run is aborted.

2. Power is calculated using the following equation:

$$\text{Power} = \text{Energy}/\text{Timestep}$$

3. Turbine Release is calculated as:

$$\text{Turbine Release} = \text{Power}/\text{Power Coefficient}$$

If Energy is not flagged as either MAX CAPACITY or BEST EFFICIENCY and the Power Coefficient is not input, the following steps are taken:

1. Power is calculated using the following equation:

$$\text{Power} = \text{Energy}/\text{Timestep}$$

2. The max Turbine Release and Power production are found for the current operating conditions.

3. If input Power is greater than the max Power for current operating conditions, and INPUT_ENERGY _ADJUST method is chosen, Turbine Release is set equal to the max Turbine Release from 2, and Power is set equal to Power from 2. The Power Coefficient is then computed as Power divided by Turbine Release.

4. Otherwise, Turbine Release is found using the Plant Power Table and the Power Coefficient is set as Power divided by Turbine Release.

5. If the Plant Power Limit is exceeded, an error is posted.

Notes on Power Plant Cap Fraction

If the Power Plant Cap Fraction is input by the user, it is necessary for the Plant Power Table to basically be scaled back to account for the operating points when the turbines are operating at less than 100%. To do this, when Turbine Release is known and Power is to be found using the Plant Power Curve, Turbine Release is divided by the Power Plant Cap Fraction. This point is then found in the Plant Power Curve for the current operating head and the Power is found using 3-D interpolation. Finally the Power is multiplied by the Power Plant Cap Fraction to get the actual Power produced for the current timestep.

If Power is known, and Turbine release is to be found in the table. Power is multiplied by the Power Plant Cap Fraction and then this point is found in the Plant Power Curve to solve for Turbine Release. Turbine Release is then divided by the Power Plant Cap Fraction to get the actual Turbine Release for the current timestep.

21.1.2.4 Plant Power Equation Release

The Plant Power Equation Release method calculates Turbine Release using the water power equation when Energy is specified. The Plant Power Equation Release method is only available if the Plant Power Equation method is selected in the Power category. Energy must be input for this method to execute. If Energy is flagged as either **MAX CAPACITY** or **BEST EFFICIENCY**, it is considered input. If Energy is flagged as **MAX CAPACITY**, Turbine Release is set to meet the Energy request at the maximum possible turbine release. If Energy is flagged as **BEST EFFICIENCY**, the run aborts because **BEST EFFICIENCY** is not supported in this method.

If Energy is flagged **UNIT VALUES** (U), and error is issued. This flag is only available with the “Unit Power Table Release” method.

Slots Specific to this Method

☛ NONE

This method first checks to see if Turbine Release is user input or set by a rule. If it is, the run aborts because both Energy and Turbine Release cannot be input.

If the Energy slot is flagged as MAX CAPACITY, the following steps are taken:

1. Set Turbine Release to be the maximum turbine release calculated by interpolating the Net Head on the Net Head Vs Max Turbine Release table.
2. Once efficiency, Plant Cap Fraction, Net Head, and Turbine Release are all known, Power is solved for using the Power Equation. The unit compatibility factor comes from balancing units and is 102.01697767 in internal RiverWare units.

$$Power = \frac{\text{Turbine Release} \times \text{Net Head} \times \text{efficiency} \times \text{Plant Cap Fraction}}{\text{Unit Compatibility Factor}}$$

If the computed Power is greater than the Plant Power Limit, the Power is reset to the Plant Power Limit. In this case, Turbine Release is re-computed using the above equation rearranged.

3. Lastly, Energy is computed as Power multiplied by the length of the timestep.

$$Energy = \text{Power} \times \text{Length of Timestep}$$

If the Energy slot is not flagged MAX CAPACITY, the following steps are taken:

When the Energy value is known (rather than flagged **Max Capacity**), the Plant Power Equation Release method uses Energy to solve for Power and Turbine Release. Power is simply Energy divided by the length of the timestep:

$$Power = \frac{Energy}{\text{Length of Timestep}}$$

Using Power, the Net Head and Turbine Release are solved for iteratively as described below:

1. If the computed Power is greater than the Plant Power Limit, the specified energy is too large. The selected method in the Input Energy adjustment category is executed. The Reduce Input Energy method reduces the energy to the maximum possible. If the None method is selected, an error will be issued that the specified energy leads to a power that is above the Plant Power Limit.

2. Turbine Release is initially assumed zero

3. Tailwater Elevation is determined via the selected Tailwater method (the “flow” variable is set to Outflow. If Turbine Release is linked it can be assumed that the Turbine Release and Spill are separated and the “flow” variable should be set to Turbine Release.)

4. Operating Head is calculated as Pool Elevation minus Tailwater Elevation

5. Net Head is calculated as Operating Head minus Head Loss

6. Turbine Release is calculated again using the Water Power equation:

$$TurbineRelease = \frac{Power \times \text{Unit Compatibility Factor}}{\text{Net Head} \times \text{Plant Efficiency}}$$

7. The calculated Turbine Release is compared to the initial Turbine Release and the process iterates until the values converge. (Note: Convergence Percentage is a general slot on power reservoirs representing the convergence in all iterative solutions-- the slot defaults to 0.0001 if not user input.)

Once converged, the Net Head is looked up on the Net Head Vs Max Turbine Release table to get the max release. If the Turbine Release is larger than the max release times the Power Plant Cap Fraction, the selected method in the Input Energy adjustment category is executed. The Reduce Input Energy method reduces the energy to the maximum possible. Otherwise, there is too much flow and an error will be issued that the energy request cannot be met.

21.1.2.5 Peak Power Equation with Off Peak Spill Release

The Peak Power Equation with Off Peak Spill Release method calculates the necessary Turbine Release, Peak Release and Peak Time using the water power equation when Energy is specified. The method is only available if the Peak Power Equation with Off Peak Spill method is selected in the Power category. Energy must be input or set by a rule for this method to execute.

SLOTS SPECIFIC TO THIS METHOD

☛ NONE

This method first checks to see if Turbine Release is user input or set by a rule. If it is, the run aborts because both Energy and Turbine Release cannot be input. When the Energy value is known, the Peak Power Equation with Off Peak Spill Release method uses Energy to solve for Turbine Release, Peak Release and Peak Time as described below:

1. Peak Release is initially set to zero.
2. Given the net head from the previous timestep (Operating Head at previous timestep minus Head Loss), the efficiency is interpolated from the Net Head vs Efficiency table. The previous Operating Head is used as an approximation so as not to introduce an additional variable in the iteration. As a result, the Tailwater Elevation at the initial timestep must be input. The net head for the initial timestep is the initial Pool Elevation minus the initial Tailwater Elevation minus Head Loss.
3. The current Tailwater Elevation is determined using the maximum of Peak Release or the current Outflow as the value in the selected Tailwater method.
4. The Operating Head is calculated as the average Pool Elevation minus the Tailwater Elevation.
5. The net head is calculated as the Operating Head minus the Head Loss.
6. Given the net head, the Generator Capacity is interpolated from the Net Head vs. Generator Capacity table. If the capacity is above the Plant Power Limit, the Generator Capacity is reset to the Plant Power Limit.
7. Peak Release is calculated according to the power equation. The unit compatibility factor comes from balancing units and the specific weight of water; it is 102.01697767 in internal RiverWare units.

$$\text{Peak Release} = \frac{\text{Generator Capacity} \times \text{Unit Compatibility Factor}}{\text{Net Head} \times \text{Efficiency}}$$

8. The new Peak Release value is compared with the previous value and the iteration (steps 3-7) continues until the value converges. (Note: Convergence Percentage is a

general slot on power reservoirs representing the convergence in all iterative solutions-- the slot defaults to 0.0001 if not input.)

Power is set equal to the Generator Capacity and Peak Time is:

$$\text{Peak Time} = \frac{Energy}{Power}$$

Turbine Release is the Peak Release average over the timestep:

$$\text{Turbine Release} = \frac{\text{Peak Release} \times \text{Peak Time}}{\text{Timestep Length}}$$

21.1.2.6 Unit Generator Power Release

The Unit Generator Power Release method is only available when Unit Generator Power is selected in the Power category. It is used to calculate the Turbine Release required to produce a given amount of Power. Energy must be input by the user for this method to execute. There are no slots specifically associated with this method.

If Energy is flagged **UNIT VALUES** (U), and error is issued. This flag is only available with the “Unit Power Table Release” method.

The Unit Generator Power Release method begins by computing the availability and power limits of each unit type. Availability and power limit values are computed as the sum of the values from the availability and power limit columns, respectively, in the Generators Available and Limit slot. A value for availability and power limit is computed for each unit type.

The efficiency of each unit type is calculated by the following equation:

$$efficiency = \frac{powerTemp}{flowTemp}$$

PowerTemp and flowTemp, both local variables, are computed from the Best Generator Power and Best Generator Flow tables, respectively, using the current Operating Head. Each unit type is then sorted in descending order based on the computed efficiency.

In order to compute the Turbine Release associated with the known Power, the method begins to add entire unit types (operating according to the best power and flow tables and beginning with the most efficient type) until the Power is exceed or all the unit types have been added. If the Power is exceeded, the last generator type is interpolated to compute the Turbine Release exactly (see equation below). However, if all the unit types have been added and the Power cannot be met, the method assumes all unit types are operating at full capacity (according to the Full Generator Flow and Full Generator Power tables). Then if the Power is exceeded, the last generator type added is interpolated to compute the Power exactly (see equation below). However, if the Power still cannot be met, an error is posted and the run is aborted because the generators are unable to produce the amount of Power specified by the user.

The interpolation equation used to calculate Power is given below:

$$\text{Turbine Release} = \text{oneLessTypeFlow} + \frac{\text{Power} - \text{oneLessTypePower}}{\text{cumulativePower} - \text{oneLessTypePower}} \bullet (\text{cumulativeFlow} - \text{oneLessTypeFlow})$$

where oneLessTypePower is the power produced from all the previous types added (excluding the most recent type added); oneLessTypeFlow is the flow through all the previous unit types (excluding the most recent type added); cumulativePower is the power produced from all the unit types added (including the most recent type); and cumulativeFlow is the flow through all the unit types added (including the most recent type).

Note: The above equation assumes the relationship between power and flow is linear regardless of the actual relationship specified in the power and flow tables. It is also interpolating over an entire type of generators.

21.1.2.7 LCR Power Release

The LCR Power Release method calculates the release from the Lower Colorado River hydropower products. The LCR Power Release method is available only when LCR Power is selected in the Power category. Energy must be input or flagged as **BEST EFFICIENCY** (Energy cannot be flagged **MAX CAPACITY** for the LCR Power method) for this method to execute. It is determined if the requested Power demand can be met. This determination is based on the maximum possible power that can be generated for a given head. If it is possible to meet the requested Power demand, the Turbine Release is set so as to produce the requested Power.

If Energy is flagged **UNIT VALUES** (U), and error is issued. This flag is only available with the “Unit Power Table Release” method.

Slots Specific to this Method

☛ None

The first step in this method is making sure the Lower Colo Power Coeffs are known. If either of these coefficients are not known, a **RiverWare**™ error is flagged and the simulation run is aborted. Then, the LCR Input Efficiency slot is checked. is not known, If it is not known, it is assumed to be 100% efficient and the LCR Input Efficiency is set to 1.0.

If Energy is flagged as **BEST EFFICIENCY**, it is calculated as the Net Energy Request plus the value of energy in the Station Energy Table corresponding to the current day of the week.

If Energy is flagged as **MAX CAPACITY**, an error is given. If Energy is not flagged as either **BEST EFFICIENCY** or **MAX CAPACITY**, it must be input by the user.

Turbine Release is calculated using the following equation:

$$\text{Turbine Release} = \left(\frac{\text{Energy}}{\text{LCR Input Efficiency}} + (1000 \times \text{Lower Colo Power Coeff \#2})\right)$$
$$\times \frac{1000}{\text{Lower Colo Power Coeff \#1} \times \text{Timestep} \times 62.4/737.5 \times \text{Operating Head} \times 35.31467 \times 3.28084}$$

where the Timestep is in hours. The constants used in the above equation are to convert the input to **RiverWare**™ standard units.

The previous equation is based on the energy calculation equation solved for Flow and corrected to standard units (see LCR Power method):

$$\text{Energy (1000 MWH)} = \left(\left(\text{Lower Colo Power Coeff \#1} \times \frac{62.4}{737.5} \times \text{Flow (Kcfs)}\right.\right.$$
$$\left.\times \text{Timestep (hours)} \times \frac{\text{Operating Head (ft)}}{1000}\right)$$
$$\left. - \text{Lower Colo Power Coeff \#2}\right) \times \text{LCR Input Efficiency}$$

where flow is in kcfs, Timestep is in hour, and Operating Head is in feet.

The correction factors used in the above equations are presented below:

$$\text{Energy (1000 MWH)} = \text{Energy (MWH)} \times 1000$$

$$\text{Flow (cfs)} = \text{Flow (Kcfs)}/1000$$

$$\text{Operating Head (m)} = \text{Operating Head (ft)}/3.28084$$

$$\text{Turbine Release (cms)} = \text{Flow (cfs)}/35.31467$$

Once Turbine Release is calculated, it is checked against the maximum allowable turbine release. A **RiverWare**™ error is flagged and the simulation run is aborted if Turbine Release exceeds the maximum allowable turbine release.

21.1.2.8 Unit Power Table Release

This method is only available if the Unit Power Table method is selected in the Power category, (See “Unit Power Table Release” on page 759.). The method **Unit Power Table Release** calculates Turbine Release when Energy is specified. If Energy is flagged as **BEST EFFICIENCY** (B) or **MAX CAPACITY** (M) or **UNIT VALUES** (U), it is considered input.

If Energy is flagged B, the Unit Best Turbine Q table will be used to determine the best efficiency Turbine Release for the current average Operating Head. This assumes that all units are in use unless specified otherwise in the Unit is Generating slot. Power is then found using the Unit Power Table. If Energy is flagged M, the Unit Max Turbine Q table is used to determine the maximum Unit Turbine Release for the current average Operating Head. This point is then found in the Unit Power Table to determine the maximum power that can be produced for this Operating Head. If Energy is flagged U, the method calculates Unit Turbine Release using table interpolation of Unit Energy on the Unit Power Table with the Unit Energy.

If Energy is input but not flagged as B, M, or U and Unit Energy is not input, the method will exit without calculating Unit Energy. If any of the values in Unit Energy are input, it will be used to determine the release and power.

METHOD DETAILS

This method will be called if Energy is input or set by a rule, which includes being flagged B, M, or U. This method will execute in the following manner.

if (Energy is flagged M)

{

If any of the Unit Energy[u] values are input or set by a rule, issue an error.

For each unit that is available (based on a non-zero value in the Unit is Generating slot), use 2D interpolation of Auto Unit Max Turbine Q table;

Set max release to a temporary local variable, Qmax[u];

Turbine Release is set to Σ Qmax[u];

Once the value for each unit flow at the current average Operating Head is found, the Unit Power[u] produced for that flow can be determined directly from the Unit Power Table.

}

else if (Energy is flagged B)

{

If any of the Unit Energy[u] values are input or set by a rule, issue an error.

For each unit that is available (based on a non-zero value in the Unit is Generating slot), use 2D interpolation of Auto Unit Best Turbine Q table to determine release at B;

Set best release to a temporary local variable, Qbest[u];

Turbine Release is set to Σ Qbest[u];

Again, Unit Power[u] will then be able to be determined directly from the Unit Power Table.

}

else if (Energy is Input/Rules (including U flag) and Unit Energy for any unit is not input/rules)

{

Issue an error; there is no way to calculate Unit Energy from plant values and no way

to calculate plant Power without unit information
}
else if (Energy is input/rules (including U flag) and Unit Energy for any unit is input/rules)
{
If Unit is Generating is set (input/rules) to 0 for a unit that has a Unit Energy, issue an error.
If Unit is Generating is set (input/rules) to 1 for a unit that does not have a Unit Energy, issue an error.
If Energy is flagged U, $Energy = \Sigma UnitEnergy[u]$; otherwise, if $Energy \neq \Sigma UnitEnergy[u]$, an error is issued
Next, Unit Power[u] = Unit Energy[u] / time (hrs)
$Power = \Sigma UnitPower[u]$
From this power calculation, the Unit Turbine Release[u] can then be determined using a reverse table lookup of Unit Power[u] in the Unit Power Table. If the Shared Penstock Head Loss method is selected, the solution is iterative as the net operating head is a function of Turbine Release. If Unit Energy[u] is less than zero, the Unit Turbine Release[u] is set to zero. A negative Unit Energy can be set to represent a unit that is spinning but not producing power (i.e. condensing).
Turbine Release = Σ Unit Turbine Release[u]
}
Finally, the method returns to the Unit Power Table method and computes Unit is Generating and Number of Units Generating. See “Unit Power Table” on page 740..

21.1.3 Power Unit Information

This category is used to provide information on unit information when one of two plant level power methods is selected. This category is only available if the **Plant Efficiency Curve** or **Plant Power Coefficient** methods are selected.. In this category are two methods:

21.1.3.1 None

This is the default, no-action method.

21.1.3.2 Plant Power Table with Units

When selected, the **Plant Power Table with Units** method allows the user to specify the number of units associated with each Turbine Release / Power combination on the Plant Power Table. In addition, at the end of the power method, the Operating Head and Turbine Release are looked up to compute the number of units that are generating.

SLOTS ASSOCIATED WITH THE METHOD:

☛ **PLANT POWER TABLE**

Type: Table Slot
Units: LENGTH, FLOW, POWER, NONE
Description: 3-D table used to determine power using interpolation
Information: Data must be entered into the table in increasing, concave blocks of the same Operating Head for the 3-dimensional table interpolation to work correctly. For every block of the same Operating Head in column 1, Turbine Release should be listed in increasing, concave order in column 2, and the corresponding Power in column 3. The number of units should be increasing in column 4. There should also be a point of zero Turbine Release and zero Power for each operating head. The second to last row for each operating head is the point of best efficiency. The last row for each operating head is the point of maximum Turbine Release and maximum Power production. If there are only two rows for a given operating head, both the **best efficiency** and **max capacity** are equal to the second row. The table shown below is an example of the proper way to formulate the Plant Power Table, with units.

Operating Head	Turbine Release	Power	Number of Units
30	0	0	0
30	100	100	1
30	200	175	2
40	0	0	0
40	100	125	1
40	220	195	2

Operating Head	Turbine Release	Power	Number of Units
50	0	0	0
50	110	147	1
50	250	205	2

I/O: Input Only

Links: Not Linkable

☛ Number of Units Generating

Type: Series
Units: NONE
Description: The number of units generating on this timestep
Information: The value in this slot is computed by looking up the Operating Head and Turbine Release on the Plant Power Table to find the number of units. Note, this computation is a 3D interpolation on the Plant Power Table so there can be a fractional number of units generating, i.e. 1.7.
I/O: Output Only

Links: Not linkable

21.1.4 Input Energy Adjustment

This method category is only available for Plant Power Coefficient Release, Plant Efficiency Curve Release, or Plant Power Equation Release methods in the Power Release category. Its purpose is to adjust input Energy values if they violate a physical constraint.

21.1.4.1 None

This is the default method. It performs no calculations and there are no slots associated with it. The Energy values will not be adjusted if this method is selected.

21.1.4.2 Reduce Input Energy

This method is used to reduce the input Energy value whenever it exceeds the maximum power (due to turbine capacity).

Slots Specific to this Method

☛ **Requested Energy**

Type: Series Slot
Units: FLOW
Description: The Energy value before being adjusted
Information: This slot is available so that the user can see when an Energy value is adjusted. The value in this slot is the energy value before being adjusted. A value exists in this slot only if the Energy value is adjusted.
I/O: Output only
Links: Not linkable

If the Energy slot value leads to a power that is greater than the maximum reservoir power (due to plant capacity, Plant Power Limit, etc), this method saves the Energy value in the Requested Energy slot. Then, the Maximum Capacity flag is set on the Energy slot. The reservoir is then forced to resolve with the Energy set to Max Capacity (instead of the original, input value). When the reservoir solves the second time, it computes the maximum reservoir Energy and sets this value on the Energy slot. The Maximum Capacity flag remains on the Energy slot for the timestep in question (and will be saved with the model file).

21.1.5 Power Plant Failure

This category is available when any of the power methods are chosen.

21.1.5.1 None

No power plant failure is modeled. If not input, the **Power Plant Cap Fraction**, if used, is set to 1.0.

21.1.5.2 Max Pool, Outflow, Tailwater

During high flow events, certain conditions cause the power plant to fail and no power can be produced. This method model the following conditions:

- Maximum pool elevation
- Maximum tailwater elevation
- Maximum outflow

Each of the above actually has two values, the first, lower value represents the shutoff criteria. The second, higher value represents the failure criteria. If the shutoff criteria is exceeded, then no power can be produced for that timestep, but if the conditions receded below the criteria, then power can again be produced. If the failure criteria is exceeded, the power plant has failed and no power can be produced from that point forward.

For example, a Pool Elevation above the shutoff limit requires the plant to cease generation. However, the power house is not flooded, and when the pool drops back below this limit, the plant can resume generation. In the second case, the pool is above the failure limit and requires the plant to cease generation, but also floods the powerhouse. In this case, even if the pool drops back below the criteria, the plant cannot resume generation, i.e the plant must "fail" for the rest of the simulation.

SLOTS ADDED BY THIS METHOD:

This method will instantiate slots in the following list.

☛ MAX POOL ELEVATION FOR POWER

Type: Table
Units: LENGTH, LENGTH
Description: Elevations at which no power can be produced
Information: This is 1X2 table slot. The first column contains the **shutoff** elevation. The second column contains a higher elevation representing the **failure** elevation.
I/O: Optional input
Links: NA

☛ MAX TAILWATER ELEVATION FOR POWER

Type: Table

Units: LENGTH, LENGTH
Description: Tailwater elevation at which power can no longer be generated
Information: This is 1X2 table slot. The first column contains the **shutoff** tailwater elevation. The second column contains a higher elevation representing the **failure** tailwater elevation. Once the failure elevation is exceeded, the plant has failed and no power can be produced on any subsequent timesteps.
I/O: Optional input
Links: NA

MAX OUTFLOW FOR POWER

Type: Table
Units: FLOW, FLOW
Description: Reservoir outflow at which power can no longer be generated
Information: This is 1X2 table slot. The first column contains the **shutoff** outflow. The second column contains a higher outflow representing the outflow at which the power plant **fails**.
I/O: Optional input
Links: NA

POWER PLANT CAP FRACTION

Type: Series
Units: NO UNITS
Description: This slot tracks whether power production is possible at this timestep
Information: A value of 1 indicates power can be generated, a value of 0 indicates no power can be generated.
I/O: Input or Output
Links: NA

METHOD DETAILS:

This method is executed at the beginning of each power method. First, the failure conditions are checked, then the shutoff conditions are checked.

Failure is tracked using the **Power Plant Cap Fraction.** If **Power Plant Cap Fraction** is 1.0 the power plant is available. If it is 0.0, the plant has failed.

At the beginning of the power method, the following logic determines the **Power Plant Cap Fraction** to use.

If the **Power Plant Cap Fraction** is input, that input value is used and no further checking will be done.

If the previous **Power Plant Cap Fraction** is 0.0, then the current **Power Plant Cap Fraction** is set to zero. This indicates the power plant has failed on previous timesteps and should remain failed. No further checking is done.

Else, the previous **Power Plant Cap Fraction** is unknown or non-zero. Then, if any of the following are true, then the **Power Plant Cap Fraction** is set to 0.0; the plant has failed.

$$\text{Pool Elevation}[t-1] > Max \text{ Elevation for Power}[failure]$$

$$\text{Tailwater Elevation}[t-1] > \text{Max TailwaterElevation for Power}[failure]$$

$$Outflow[t-1] > Max \text{ Outflow for Power}[failure]$$

A diagnostic is available in the User Methods category that describes any failure constraints in effect.

Otherwise, the **Power Plant Cap Fraction** is set to 1.0

Even if the plant “fails” in the course of a simulation, the user can “restart” it manually by setting a non-zero value in the **Power Plant Cap Fraction.**

If any of the following are true, then the plant has failed or shutoff. Turbine Release, Power, Energy and a few method specific slots (as described in the power method section) are set to zero.

$$\text{Power Plant Cap Fraction}[t] = 0.0$$

$$\text{Pool Elevation}[t-1] > Max \text{ Elevation for Power}[shutoff]$$

$$\text{Tailwater Elevation}[t-1] > \text{Max TailwaterElevation for Power}[shutoff]$$

$$Outflow[t-1] > Max \text{ Outflow for Power}[shutoff]$$

$$\text{Pool Elevation}[t-1] < \text{Minimum Power Elevation}$$

Note, the last equation is the behavior for minimum power pool. This slot is added by some of the power methods, not this method.

Also note, if Power Plant Cap Fraction is not zero but one of the other constraints is true, the power slots are set to zero, but the plant has not failed, so future timesteps can generate power.

If none of the above are true, the power method then proceeds as before using the computed **Power Plant Cap Fraction**.

21.1.6 Energy in Storage

The methods available in the Energy in Storage category are used to calculate the total energy that could be produced by the water stored in the reservoir.

21.1.6.1 None

Chosen if the user does not want to calculate the Energy In Storage. No slots are specifically associated with this method. This method performs no calculations.

21.1.6.2 EIS Table Lookup

The EIS TableLookup method obtains the amount of Energy In Storage from a table of Pool Elevation vs. Energy In Storage values and the Pool Elevation.

Slots Specific to this Method

☛ **Energy In Storage**

Type: AggSeriesSlot
Units: ENERGY
Description: Energy In Storage in the Reservoir
Information:
I/O: Output only
Links: Usually not linked

☛ **Energy In Storage Table**

Type: TableSlot
Units: LENGTH vs. ENERGY
Description: Pool Elevation vs. Energy In Storage In the Reservoir
Information:
I/O: Required input
Links: Not linkable

The calculations involved with this method are very simple. The Pool Elevation for the current timestep is used to determine the Energy In Storage using the Energy In Storage Table. Simple linear interpolation is used.

21.1.6.3 EIS Table Lookup with Cons Pool

This method will be available only when the **Conservation Pool** or **Conservation and Flood Pools** method in the **Operating Levels** category is selected.

Slots Specific to this Method

☛ **Energy In Storage**

Type: AggSeriesSlot
Units: ENERGY
Description: Energy In Storage in the Reservoir

Information:
I/O: Output only
Links: Usually not linked

☛ Energy In Storage Table
Type: TableSlot
Units: LENGTH vs. ENERGY
Description: Pool Elevation vs. Energy In Storage In the Reservoir
Information:
I/O: Required input
Links: Not linkable

☛ Conservation Pool Full EIS
Type: Series Slot
Units: MWH
Description: The EIS at the Top of the Conservation Pool
Information:
I/O: Output Only
Links: Not Linkable

The method is executed at the end of each dispatch method.

The **Pool Elevation** for the timestep is looked up in **Energy In Storage Table**. Simple linear interpolation is used. The resulting **Energy in Storage** is then set on the slot.

Next, the **Conservation Pool Full EIS** is calculated as follows:

$$\text{Top Conservation Pool Elevation}[t] = \text{Operating Level Table}[t, \text{Top of Conservation Pool Level}]$$

$$\text{Conservation Pool Full EIS}[t] = \text{Energy in Storage Table}[\text{Top of Conservation Pool Elevation}[t]]$$

Note, in the equation, the appropriate Operating Level Table will be used based on the timestep and the computation in the selected method in the **Conditional Operating Levels** category.

21.1.7 Tailwater

The Tailwater methods (available in the Tailwater category of the User Selectable Methods) calculate the Tailwater Elevation of a Level Power Reservoir. The Tailwater Elevation represents the water surface elevation immediately downstream of the Power Plant. This parameter is required to calculate the Operating Head of the Power Reservoir which is used to calculate either the Energy or Turbine Release of that Reservoir. The Tailwater methods are dependent upon the Reservoir being a Pumped Storage Reservoir and a valid Power method (available in the Power category of the User Selectable Methods) being selected. These methods require a valid Outflow to perform their calculations.

21.1.7.1 None

This is the default method in the Tailwater category. This method performs no calculations. There are no slots specifically associated with this method.

Slots Specific to this Method

☛ **None**

21.1.7.2 Linked or Input

The Linked or Input method allows the user to either input values for the Tailwater Elevation or link the Tailwater Base Value to a slot (Pool Elevation or Backwater Elevation) on a downstream Reservoir. If the Tailwater Base Value is linked, the Tailwater Elevation cannot be input by the user. If the Tailwater Base Value is not linked, the Tailwater Elevation must be input by the user. The Tailwater Elevation is determined by following procedure if it is not input (i.e. the Tailwater Base Value is linked).

Slots Specific to this Method

☛ **Tailwater Base Value**

Type: Series
Units: LENGTH
Description: elevation of tailwater or base elevation used to compute elevation of tailwater
Information:
I/O: Optional; may be input or set by a link.
Links: May be linked to the Pool Elevation or Backwater Elevation of a downstream Reservoir.

The Linked or Input user method does not perform any calculations per se. There are however many logical evaluations performed in this method. The Linked or Input method is discussed below.

If a value for the Tailwater Base Value has been calculated by another user method or propagated through a link, the temporary value for the Tailwater Base Value (TWBaseValueTemp) is set equal to the calculated or linked Tailwater Base Value.

If the Tailwater Base Value is linked to another slot, the following steps are performed:

1. If the Tailwater Elevation is input, a **RiverWare**™ error is flagged and the run is aborted.

2. If TWBaseValueTemp is known, the Tailwater Elevation is set equal to the TWBaseValueTemp.

3. If TWBaseValueTemp is not known, but the previous timestep's Tailwater Elevation is known, the Tailwater Elevation is set equal to the previous timestep's Tailwater Elevation.

4. If neither the TWBaseValueTemp nor the previous timestep's Tailwater Elevation is known and the previous timestep was the initial timestep of the simulation, a **RiverWare**™ error is flagged and the simulation run is aborted.

5. If neither the TWBaseValueTemp nor the previous timestep's Tailwater Elevation is known and the previous timestep was not the initial timestep, the method is exited and waits for more information.

If the Tailwater Base Value is not linked to another slot, Tailwater Elevation must be input. If the Tailwater Elevation is not input, a **RiverWare**™ error is flagged and the simulation run is aborted.

21.1.7.3 Base Value Only

The Base Value Only method is similar to the Linked or Input method. If this method is selected, the user must either input values directly into the Tailwater Base Value slot or link the Tailwater Base Value slot to either the Pool Elevation slot or the Backwater Elevation slot of the downstream Reservoir. Either the initial Tailwater Elevation or the initial Tailwater Base Value must be input by the user for this method to execute successfully. The Tailwater Elevation for any timestep is computed as the average Tailwater Base Value over the timestep.

SLOTS SPECIFIC TO THIS METHOD

☛ **Tailwater Base Value**

Type: Series
Units: LENGTH
Description: elevation of tailwater or base elevation of tailwater is used to compute the tailwater elevation
Information:
I/O: Optional; may be input or linked.
Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

This method is based on logic similar to that of the Linked or Input method. The Base Value Only user method performs many logical evaluations to set the Tailwater Elevation. This method is described below.

The temporary Tailwater Base Value (TWBaseValueTemp) is set equal to the Tailwater Base Value if the Tailwater Base Value was calculated by another user method or set via a link.

If the Tailwater Base Value is linked to another slot, the following steps are performed:

1. If the previous timestep's Tailwater Base Value is not known, and both the previous timestep's Tailwater Elevation and TWBaseValueTemp are known, the Tailwater Elevation is calculated using the following formula:

$$\text{Tailwater Elevation} = \frac{(\text{TWBaseValueTemp} + \text{Tailwater Elevation}(-1))}{2}$$

where: Tailwater Elevation (-1) is the previous timestep's Tailwater Elevation.

2. If neither the previous timestep's Tailwater Base Value nor the TWBaseValueTemp is known but the previous timestep's Tailwater Elevation is known, the current timestep is set equal to the previous timestep's Tailwater Elevation.

3. If neither the previous timestep's Tailwater Base Value nor the previous timestep's Tailwater Elevation is known and the previous timestep is the initial timestep, a **RiverWare**™ error is posted and the simulation run is aborted.

4. If neither the previous timestep's Tailwater Base Value nor the previous timestep's Tailwater Elevation is known and the previous timestep is not the initial timestep, more information must be known. The method is exited and waits for more information to be known.

5. If both the previous timestep's Tailwater Base Value and the TWBaseValueTemp are known, the Tailwater Elevation is calculated using the following equation:

$$\text{Tailwater Elevation} = \frac{(\text{TWBaseValueTemp} + \text{Tailwater Base Value}(-1))}{2}$$

6. If the previous timestep's Tailwater Base Value is known but TWBaseValueTemp is not known, Tailwater Elevation is set equal to the previous timestep's Tailwater Base Value.

If the Tailwater Base Value is not linked, the following steps are performed:

1. If the Tailwater Elevation is not known and the TWBaseValueTemp is known, the Tailwater Elevation is set equal to the TWBaseValueTemp.

2. If neither the Tailwater Elevation nor the TWBaseValueTemp are known, a **RiverWare**™ error is flagged and the simulation run is aborted.

21.1.7.4 Base Value Plus Lookup Table

The Base Value Plus Lookup Table method computes the Tailwater Elevation. This is done by adding the average Tailwater Base Value (over the timestep) to a function of Outflow determined by the Tailwater Table. The Outflow value used to find the corresponding Tailwater value on the Tailwater Table is either the value of the local variable Flow or the value of the Outflow slot. The Tailwater Base Value may be input by the user or linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir. If the Tailwater Base Value is neither input nor linked, it is assumed to be zero.

Slots Specific to this Method

☛ **Tailwater Base Value**

Type: Series
Units: LENGTH
Description: the base elevation of the tailwater
Information:
I/O: Optional; can be input or linked.
Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

☛ **Tailwater Table**

Type: Table
Units: FLOW VS. LENGTH
Description: reservoir outflow vs. either the tailwater elevation or the tailwater elevation increment
Information: If the Tailwater Base Value is non-zero, the Tailwater Table gives values of incremental increase in Tailwater Elevation over the Base value. Otherwise, the table gives the Tailwater Elevation values.
I/O: Required input
Links: Not linkable

This method is based on logic similar to that of the Linked or Input method. The Base Value Plus Lookup Table user method performs many logical evaluations to set the Tailwater Elevation. This method is described below.

The temporary Tailwater Base Value (TWBaseValueTemp) is set equal to the Tailwater Base Value if the Tailwater Base Value was calculated by another user method or set via a link.

If the Local Info variable Flow is known, the local variable, tempflow, is set equal to Flow. If the Local Info variable Flow is not known, but the Outflow is known, tempflow is set equal to Outflow.

If the tempflow value is known, the following steps are taken to determine the Tailwater Elevation.

If the Tailwater Base Value is linked (and tempflow is known), the following steps are performed:

1. If both TWBaseValueTemp and the previous timestep's Tailwater Base Value are known, TWTemp (a local variable) is obtained from a table interpolation performed on the Tailwater Table using tempflow. The Tailwater Elevation is then calculated using the following equation.

$$\text{Tailwater Elevation} = (\text{TWBaseValueTemp} + \text{Tailwater Base Value (-1)})/2 + \text{TWTemp}$$

2. If TWBaseValueTemp is known but the previous timestep's Tailwater Base Value is known, but the previous timestep's Tailwater Basic Value is not known. TWTemp is obtained from a table interpolation performed on the Tailwater Table using tempflow. The Tailwater Elevation is then calculated using the following equation:

$$\text{Tailwater Elevation} = \text{TWBaseValueTemp} + \text{TWTemp}$$

3. If TWBaseValueTemp is not known but the previous timestep's Tailwater Base Value is known, TWTemp is obtained from a table interpolation performed on the Tailwater Table using tempflow. The Tailwater Elevation is then calculated using the following equation:

$$\text{Tailwater Elevation} = \text{Tailwater Base Value(-1)} + \text{TWTemp}$$

4. If neither TWBaseValueTemp nor the previous timestep's Tailwater Base Value are known but the previous timestep's Tailwater Elevation is known, the current timestep's Tailwater Elevation is set equal to the previous timestep's Tailwater Elevation.

5. If neither TWBaseValueTemp nor the previous timestep's Tailwater Base Value is known and the previous timestep is the initial timestep of the run, a **RiverWare**™ error is posted and the run is aborted.

6. If neither the TWBaseValueTemp nor the previous timestep's Tailwater Base Value is known and the previous timestep is not the initial timestep of the run, the method is exited and waits for more information.

If the Tailwater Base Value is NOT linked (and tempflow is known), the following steps are performed:

1. If TWBaseValueTemp is known, TWTemp is obtained from a table interpolation performed on the Tailwater Table using tempflow. The Tailwater Elevation is then calculated using the following equation:

$$\text{Tailwater Elevation} = \text{TWBaseValueTemp} + \text{TWTemp}$$

2. If TWBaseValueTemp is not known, the Tailwater Elevation is obtained from a table interpolation performed on the Tailwater Table using tempflow.

The only case where tempflow is not known is when Outflow is not known. This scenario only occurs at the first timestep in one of the three “given Energy” dispatch methods. The following steps are performed to determine the Tailwater Elevation if tempflow is not known.

If the Tailwater Base Value is linked (and tempflow is not known), the following steps are taken:

1. If TWBaseValueTemp and the previous timestep’s Tailwater Elevation are known, the Tailwater Elevation is calculated using the following equation:

$$\text{Tailwater Elevation} = (\text{TWBaseValueTemp} + \text{Tailwater Elevation (-1)})/2$$

2. If both TWBaseValueTemp and the previous timestep’s Tailwater Base Value are known but the previous timestep’s Tailwater Elevation is not known, Tailwater Elevation is calculated using the following equation:

$$\text{Tailwater Elevation} = (\text{TWBaseValue Temp} + \text{Tailwater Base Value (-1)})/2$$

3. If TWBaseValueTemp is known, but neither the previous timestep’s Tailwater Elevation nor the previous timestep’s Tailwater Base Value are known, Tailwater Elevation is set equal to TWBaseValueTemp.

4. If TWBaseValueTemp is not known but the previous timestep’s Tailwater Elevation is known, the current timestep’s Tailwater Elevation is set equal to the previous timestep’s Tailwater Elevation.

5. If neither TWBaseValueTemp nor the previous timestep’s Tailwater Elevation are known, but the previous timestep’s Tailwater Base Value is known, Tailwater Elevation is set equal to the previous timestep’s Tailwater BaseValue.

6. If the previous timestep is the initial timestep of the run, and none of the following are known: TWBaseValueTemp, the previous timestep’s Tailwater Base Value, and the previous timestep’s Tailwater Elevation, a **RiverWare**™ error is posted and the simulation run is aborted.

7. If the previous timestep is NOT the initial timestep of the run, and none of the following are known: TWBaseValueTemp, the previous timestep’s Tailwater Base Value, and the previous timestep’s Tailwater Elevation, the method is exited and waits for more information to execute.

If the Tailwater Base Value is not linked (and tempflow is NOT known), the following procedures are performed:

1. If the Tailwater Elevation is known, the method is exited because no calculations need to be performed.

2. If the Tailwater Elevation is NOT known, and TWBaseValueTemp is NOT equal to zero, Tailwater Elevation is set equal to TWBaseValueTemp.

3. If the Tailwater Elevation is NOT known and either the temporary Tailwater Base Value is NOT known or equal to zero or both, Tailwater Elevation is set equal to the elevation corresponding to zero on the Tailwater Table.

Definitions of some of the terms used in the above equations are located below:

- ***TWBaseValueTemp -*** a temporary value for the Tailwater Base Value. This value may be determined from another method or propagated across a link.
- ***tempflow -*** a local variable used as the outflow from the Reservoir.
- ***TWTemp -*** a local variable used to represent the incremental increase in the Tailwater Elevation over the Tailwater Base Value.
- ***Tailwater Elevation (-1) -*** the Tailwater Elevation at the previous timestep.
- ***Tailwater Base Value (-1) -*** the Tailwater Base Value at the previous timestep.

21.1.7.5 Stage Flow Lookup Table

The TWstageFlowLookupTable method is similar to the other methods for determining Tailwater Elevation. The Tailwater Elevation is obtained from a 3-dimensional table relating Outflow, Downstream Stage, and the corresponding Tailwater Elevation for most cases. The data in this table must be input by the user. The value for Downstream Stage that is used in this method is the larger of the Tailwater Reference Elevation or the Tailwater Base Value if the Tailwater Base Value is linked to a downstream elevation. If the Tailwater Base Value is not linked, the temporary Tailwater Base Value is used as the Downstream Stage. An average value for the Tailwater Elevation over the timestep is used whenever possible.

SLOTS SPECIFIC TO THIS METHOD

☛ **Stage Flow Tailwater Table**

Type: Table
Units: FLOW VS. LENGTH VS. LENGTH
Description: Reservoir Outflow vs. Downstream Elevation (Tailwater Base Value) vs. Tailwater Elevation
Information: Data must be entered into the table in increasing blocks of the same Outflow value for the 3-dimensional table interpolator to work correctly. For every

block of same Outflows in column 1, Stages should be listed in increasing order in column 2, and the corresponding Tailwater Elevations in column 3.

Outflow	Downstream Stage	TW Elevation
100	500	510
100	550	560
100	600	610
200	500	520
200	550	570
200	600	620
300	500	530
300	550	580
300	600	630

I/O: Required input

Links: Not linkable

☛ **Tailwater Base Value**

Type: Series
Units: LENGTH
Description: base elevation of the tailwater
Information:
I/O: Optional; can be input or set by a link.

Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

☛ **Tailwater Reference Elevation**

Type: Table
Units: LENGTH
Description: lowest Reservoir discharge Elevation when there are no backwater effects from a downstream pool (reservoir)
Information: If this slot has input data, the greater of the Tailwater Reference Elevation or the linked lower reservoir’s Pool or Backwater Elevation is used to calculate the Tailwater Base Value. If the Tailwater Base Value is linked to a downstream elevation, this value must be specified by the user. If the Tailwater Base Value is not linked, this value is not used, even if specified.
I/O: Required input

Links: Not linkable

The TWstageFlowLookupTable user method performs many logical tests to determine the Tailwater Elevation. This method is described below.

The first step in this method is to check and see if the Tailwater Reference Elevation is known. If the Tailwater Reference Elevation is not known, a **RiverWare**™ error is posted and the run is aborted.

The temporary Tailwater Base Value (TWBaseValueTemp) is set equal to the Tailwater Base Value if the Tailwater Base Value was calculated by another user method or set via a link.

If the local variable Flow is known, the local variable tempflow is set equal to Flow If the local variable Flow is not known, but the Outflow is known, if the Local Info variable Flow is not known but Outflow is known, tempflow is set equal to Outflow.

If the tempflow value is known, the following steps are taken to determine the Tailwater Elevation.

If the Tailwater Base Value is linked (and tempflow is known), the following steps are performed:

1. If both TWBaseValueTemp and the previous timestep's Tailwater Base Value are known, downstreamStage is computed as the average of the previous timestep's Tailwater Base Value and either the Tail Water Reference Elevation or TWBaseValueTemp (whichever is greater). A table interpolation is performed to determine the Tailwater Elevation using the downstreamStage, tempflow, and the Stage Flow Tailwater Table.

2. If TWBaseValueTemp is known but the previous timestep's Tailwater Base Value is NOT known, downstreamStage is set equal to either the Tail Water Reference Elevation or TWBaseValueTemp (whichever is greater). A table interpolation is performed to determine the Tailwater Elevation using the downstreamStage, tempflow, and the Stage Flow Tailwater Table.

3. If the previous timestep's Tailwater Base Value is known but TWBaseValueTemp is NOT known, downstreamStage is set equal to the previous timestep's Tailwater Base Value. A table interpolation is performed to determine the Tailwater Elevation using the downstreamStage, tempflow, and the Stage Flow Tailwater Table.

4. If neither TWBaseValueTemp nor the previous timestep's Tailwater Base Value are known, but the previous timestep's Tailwater Elevation is known, the current timestep's Tailwater Elevation is set equal to the previous timestep's Tailwater Elevation.

5. If the previous timestep was the initial timestep and none of the following are known: TWBaseValueTemp, the previous timestep's Tailwater Base Value, and the previous timestep's Tailwater Elevation, a **RiverWare**™ error is posted and the run is aborted.

6. If the previous timestep was NOT the initial timestep, and none of the following are known: the temporary TWBaseValueTemp, the previous timestep's Tailwater Base Value, and the previous timestep's Tailwater Elevation, the method is exited and waits for more information.

If the Tailwater Base Value is not linked (and tempflow is known), the following steps are performed:

1. If TWBaseValueTemp is known, downstreamStage is set equal to TWBaseValueTemp. A table interpolation is performed to determine the Tailwater Elevation using the downstreamStage, tempflow, and the Stage Flow Tailwater Table.

2. If TWBaseValueTemp is NOT known, a **RiverWare**™ error is posted and the run is aborted.

The only case where tempflow is not known is when Outflow is not known. This scenario only occurs at the first timestep in one of the three “given Energy” dispatch methods. The following steps are performed to determine the Tailwater Elevation if tempflow is not known.

If the Tailwater Base Value is linked (and tempflow is not known), the following steps are performed:

1. If both TWBaseValueTemp and the previous timestep’s Tailwater Elevation are known, the Tailwater Elevation is calculated using the following equation:

$$\text{Tailwater Elevation} = (\text{TWBaseValue Temp} + \text{Tailwater Elevation}(-1))/2$$

2. If both TWBaseValueTemp and the previous timestep’s Tailwater Base Value are known but the previous timestep’s Tailwater Elevation is not known, Tailwater Elevation is computed using the following equation:

$$\text{Tailwater Elevation} = (\text{TWBaseValue Temp} + \text{Tailwater Base Value}(-1))/2$$

3. If TWBaseValueTemp is known but neither the previous timestep’s Tailwater Elevation nor the previous timestep’s Tailwater Base Value are known, Tailwater Elevation is set equal to TWBaseValueTemp.

4. If TWBaseValueTemp is not known but the previous timestep’s Tailwater Elevation is known, the current timestep’s Tailwater Elevation is set equal to the previous timestep’s Tailwater Elevation.

5. If neither TWBaseValueTemp nor the previous timestep’s Tailwater Elevation is known, but the previous timestep’s Tailwater Base Value is known, Tailwater Elevation is set equal to the previous timestep’s Tailwater Base Value.

6. If the previous timestep is the initial timestep of the run, and none of the following are known: TWBaseValueTemp, the previous timestep’s Tailwater Base Value, and the previous timestep’s Tailwater Elevation, a **RiverWare**™ error is posted, and the run is aborted.

7. If the previous timestep is NOT the initial timestep of the run and none of the following are known: TWBaseValueTemp, the previous timestep's Tailwater Base Value, and the previous timestep's Tailwater Elevation, the method is exited and waits for more information.

If the Tailwater Base Value is not linked (and tempflow is not known) the following procedures are performed:

1. If the Tailwater Elevation is known, the method is exited because no calculations need to be performed.

2. If the Tailwater Elevation is NOT known and the TWBaseValueTemp is NOT equal to zero, Tailwater Elevation is set equal to TWBaseValueTemp.

3. If the Tailwater Elevation is NOT known and either the temporary Tailwater Base Value is NOT known or equal to zero or both, Tailwater Elevation is set equal to the elevation corresponding to zero on the Tailwater Table.

Definitions of some of the terms used in the above equations are located below:

- ***TWBaseValueTemp* -** a temporary value for the Tailwater Base Value. This value may be determined from another method or propagated across a link.
- ***tempflow* -** a local variable used as the outflow from the Reservoir.
- ***downstreamStage* -** a local variable used to hold the value of the downstream stage elevation.
- ***Tailwater Elevation (-1)* -** the Tailwater Elevation at the previous timestep.
- ***Tailwater Base Value (-1)* -** the Tailwater Base Value at the previous timestep.

21.1.7.6 Compare to Avg Base Value

The Compare to Avg Base Value method uses similar methodology as the Base Value Plus Lookup Table and TWstageFlowLookupTable methods. The only difference between this method and the others is that a new local variable, TWCompare, is used. The value for TWCompare is compared with a table lookup value for the Tailwater Elevation. The larger of the two values is used as the Tailwater Elevation. An average Tailwater Elevation over the timestep is calculated whenever possible.

Slots Specific to this Method

Tailwater Base Value

Type: Series
Units: LENGTH
Description: base elevation of the tailwater
Information:
I/O: Optional; can be input of set by a link.

Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

☛ Tailwater Table

Type: Table
Units: FLOW vs. LENGTH
Description: Reservoir Outflow vs. either the Tailwater Elevation or the tailwater elevation increment
Information: If the Tailwater Base Value is non-zero, the Tailwater Table holds values of incremental increase in Tailwater Elevation over the Base value. Otherwise, the table holds the Tailwater Elevation values.
I/O: Required input
Links: Not linkable

The Compare to Avg Base Value user method performs many logical evaluations to determine the Tailwater Elevation. This method is described below.

The temporary Tailwater Base Value (TWBaseValueTemp) is set equal to the Tailwater Base Value is the Tailwater was calculated by another user method or set via a link. For the Tailwater Compare Method, the Tailwater Base Value must be linked to another object. Otherwise there would be no reason to compare values. If the Tailwater Base Value is not linked, a **RiverWare**™ error is posted and the simulation run is aborted.

If the Local Info variable Flow is known, the local variable tempflow is set equal to Flow.

If the local variable Flow is not known, but Outflow is known, tempflow is set equal to Outflow.

If the tempflow value is known, the following steps are taken to determine the Tailwater Elevation.

1. If both TWBaseValueTemp and the previous timestep's Tailwater Base Value are known,

$$\text{TWCompare} = (\text{TWBaseValueTemp} + \text{Tailwater Base Value}(-1))/2$$

2. If TWBaseValueTemp is known but the previous timestep's Tailwater Base Value is NOT known, TWCompare is set equal to TWBaseValueTemp.

3. If TWBaseValueTemp is not known but the previous timestep's Tailwater Base Value is known, TWCompare is set equal to the previous timestep's Tailwater Base Value.

4. If neither TWBaseValueTemp nor the previous timestep's Tailwater Base Value are known and the previous timestep is the initial timestep, a **RiverWare**™ error is posted and the run is aborted.

5. If neither TWBaseValueTemp nor the previous timestep's Tailwater Base Value are known and the previous timestep is NOT the initial timestep, the method is exited and waits for more information.

Once the TWCompare value has been determined, a table interpolation is performed using the Tailwater Table to obtain the tailwater elevation corresponding to the tempflow value. The tailwater elevation value obtained from this interpolation is then compared to TWCompare. The largest of the two values is used to set the Tailwater Elevation.

The only case where tempflow is not known is when Outflow is not known. This scenario only occurs at the first timestep in one of the three "given Energy" dispatch methods. The following steps are performed to determine the Tailwater Elevation if tempflow is not known.

1. If both TWBaseValueTemp and the previous timestep's Tailwater Elevation are known,

$$\text{Tailwater Elevation} = (\text{TWBaseValue Temp} + \text{Tailwater Elevation (-1)})/2$$

2. If both TWBaseValueTemp and the previous timestep's Tailwater Base Value are known but the previous timestep's Tailwater Elevation is not known,

$$\text{Tailwater Elevation} = (\text{TWBaseValue Temp} + \text{Tailwater Base Value (-1)})/2$$

3. If TWBaseValueTemp is linked and the temporary Tailwater Base Value is known but neither the pervious timestep's Tailwater Elevation nor the previous timestep's Tailwater Base Value are known, the Tailwater Elevation is set equal to TWBaseValueTemp.

4. If TWBaseValueTemp is not known but the previous timestep's Tailwater Elevation is known, the current timestep's Tailwater Elevation is set equal to the previous timestep's Tailwater Elevation.

5. If neither TWBaseValueTemp nor the previous timestep's Tailwater Elevation are known, but the previous timestep's Tailwater Base Value is known, the current timestep's Tailwater Elevation is set equal to the previous timestep's Tailwater Base Value.

6. If the previous timestep is the initial timestep of the run, and none of the following are known: TWBaseValueTemp, the previous timestep's Tailwater Base Value, and the previous timestep's Tailwater Elevation, a **RiverWare**™ error is posted and the simulation is aborted.

7. If the previous timestep is NOT the initial timestep of the run and none of the following are known: TWBaseValueTemp, the previous timestep's Tailwater Base Value, and the previous timestep's Tailwater Elevation, the method is exited and waits for more information.

Definitions of some of the terms used in the above equations are located below:

- **TWBaseValueTemp**: a temporary value for the Tailwater Base Value. This value may be determined from another method or propagated across a link.
- **tempflow**: a local variable used as the outflow from the Reservoir.
- **TWCompare**: a local variable used to hold a value that is compared to the Tailwater Elevation obtained by table interpolation.
- **Tailwater Elevation (-1)**: the Tailwater Elevation at the previous timestep.
- **Tailwater Base Value (-1)**: the Tailwater Base Value at the previous timestep.

21.1.7.7 Coefficients Table

The Coefficients Table method multiplies Outflow, Tailwater Base Value, and Tailwater Elevation at the current and/or previous timestep by coefficients that are stored on a table. These products are added together along with constants to compute the Tailwater Elevation.

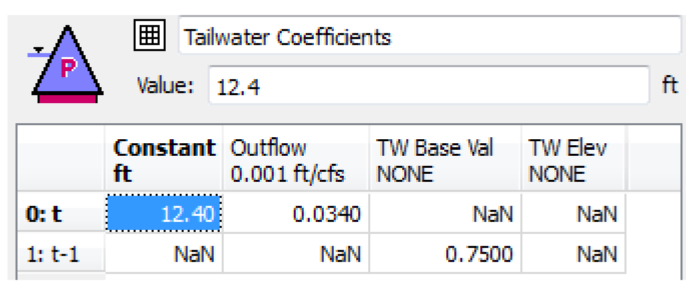
SLOTS SPECIFIC TO THIS METHOD

☛ TAILWATER BASE VALUE

Type: Series
Units: LENGTH
Description: base elevation of the tailwater
Information:
I/O: Optional; can be input or set by a link.
Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

☛ TAILWATER COEFFICIENTS

Type: Table
Units: LENGTH, LENGTH/FLOW, NONE, NONE
Description: This table contains the coefficients used in the calculation. The columns are as follows: **Constant**, **Outflow**, **TW Base Val**, **TW Elev**. The first row (**t**) represents the value to multiply by the current timestep's value. The second row (**t-1**)represents the value to multiply by the previous timestep's value.



	Constant ft	Outflow 0.001 ft/cfs	TW Base Val NONE	TW Elev NONE
0: t	12.40	0.0340	NaN	NaN
1: t-1	NaN	NaN	0.7500	NaN

Information: Not every cell will have a value, but there must be at least one value in the table. In addition, the coefficient for **TW Elev** at t cannot be non-zero or an error will be issued. (Tailwater Elevation at the current timestep is the value being computed, so it is not possible to use it in the calculation). If any coefficents are specified (non-zero) for t-1 for **Outflow**, **TW Base Val**, or **TW Elev**, then the initial timestep value of those slots must also be specified.
I/O: Input only
Links: Not Linkable

At the beginning of the run, the **Tailwater Coefficients** table is verified for valid coefficients as described in the slot information above. Not every coefficient has to be specified.

When the tailwater method is executed, the following computation is performed.

$$\begin{aligned}\text{Tailwater Elevation} = {} & \text{Constant Coeff}[t] + \text{Constant Coeff}[t-1] + \\ & \text{Outflow Coeff}[t] \times \text{flow} + \text{Outflow Coeff}[t-1] \times \text{Outflow}[t-1] + \\ & \text{TW Base Val Coeff}[t] \times \text{TailwaterBaseValueTemp}[t] + \\ & \text{TW Base Val Coeff}[t-1] \times \text{Tailwater BaseValue}[t-1] + \\ & \text{TW Elev Coeff}[t-1] \times \text{Tailwater Elevation}[t-1]\end{aligned}$$

Each coefficient and each series value are checked for validity before including in the total.

In the above equation, the values for **flow** and **TailwaterBaseValueTemp** are based on the information that is known:

- The *flow* variable represents the outflow passed into the method if valid, or the value on the Outflow slot if not passed in. If neither is valid, Tailwater Elevation is set to the previous Tailwater Elevation. If that is not valid, the method exits and waits for a flow value to become valid.

The *TailwaterBaseValueTemp* is either the base value passed into the method or the value on the Tailwater Base Value slot. If neither is known, the previous Tailwater Base Value is used. If that is not valid, the method executes to wait for more information.

21.1.7.8 Hoover Tailwater

The Hoover Tailwater method was developed by the USBR for use on Lake Mead. This method uses a table of empirically derived constants.

Slots Specific to this Method

☛ **Tailwater Base Value**
Type: Series
Units: LENGTH
Description: base elevation of the tailwater
Information:
I/O: Optional; can be input or set by a link.

Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

Tailwater Table

Type: Table
Units: FLOW vs. LENGTH
Description: Reservoir Outflow vs. either the Tailwater Elevation or the tailwater elevation increment
Information: If the Tailwater Base Value is non-zero, the Tailwater Table holds values of incremental increase in Tailwater Elevation over the Base value. Otherwise, the table holds the Tailwater Elevation values.
I/O: Required input
Links: Not linkable

Hoover Tailwater Table

Type: Table
Units: NO UNITS
Description: constants for equation that calculates the Tailwater Elevation
Information: This method was developed by the USBR for use on Lake Mead. Column 1 is the TWaverage in feet, column 2 is hte5, column 3 is coeff1.
I/O: Required input
Links: Not linkable

The Hoover Tailwater user method performs calculations to determine the Tailwater Elevation. This method is described below:

The first step of this method is to set the temporary Tailwater Base Value (TWBaseValueTemp). TWBaseValueTemp is set equal to the Tailwater Base Value if the Tailwater Base Value was calculated by another user method or set via a link.

Then the previous timestep's Tailwater Base Value is checked. If the previous timestep's Tailwater Base Value is not known, a **RiverWare**™ error is posted and the run is aborted.

The TWaverage, a local variable, is then calculated using the following steps:

1. If TWBaseValueTemp is known,

$$\text{TWaverage} = (\text{TWBaseValueTemp} + \text{Tailwater Base Value}(-1))/2$$

2. If the temporary TWBaseValueTemp is NOT known, the TWaverage is set equal to the previous timestep's Tailwater Base Value.

Once the TWaverage has been calculated, it is used to interpolate the empirical constants for the following equation from the Hoover Tailwater Table.

Finally, $\text{TELEFT} = \text{hte5} + ((\text{Qcfs})/1000 - 5) \times \text{coeff1}$

Tailwater Elevation is then set equal to TELEFT (which is converted into internal units).

Definitions of some of the terms used in the above equations are located below:

- **TWBaseValueTemp**: a temporary value for the Tailwater Base Value. This value may be determined from another method or propagated across a link.
- **Tailwater Base Value (-1)**: the Tailwater Base Value at the previous timestep.
- **TWaverage**: a local variable that is used as the average of the Tailwater Elevation over the timestep.
- **TELEFT**: the Tailwater Elevation in feet as determined by the equation given above.
- **hte5**: an empirical constant developed by the USBR.
- **Qcfs**: the flow in cfs.
- **coeff1**: an empirical constant developed by the USBR.

21.1.8 Pump Power

Used to calculated flow, power, and pump efficiency.

21.1.8.1 None

This is the default method. It performs no calculations and has no slots associated with it. This method is not a valid method and an error is flagged if the user has selected it.

21.1.8.2 Unit Pump Power

Calculates pump power and pumped flow. First, the efficiency of all the different pump types is determined. Then flow and power are calculated based on the properties that are given. There are no slots specifically associated with this method.

The first step in the method is to determine the Operating Head. Operating head is calculated by the user method selected in the Tailwater category. Then, the efficiencies of all pump types at the given Operating Head are calculated. Efficiency is calculated as power divided by flow where power and flow are determined from interpolation of the Head vs Pump Power and Head vs Pump Flow tables, respectively, for each pump type.

To solve for Pump Power and Inflow, different sets of calculations are performed depending upon the slots input by the user.

If Pumps Used is given, the following calculations are performed (pumps are added in the best efficiency order):

1. For each pump used, the flow and power are calculated from the Head vs Pump Flow and Head vs Pump Power tables.
2. Pumped Flow is set as the sum of the flows through each pump. Pump Power is set as the sum of the power used by each pump.
3. Pump Energy is calculated as the product of Pump Power and the length of the timestep.

If Pumped Flow is given, the following calculations are performed:

1. Pump types are added (in the order of most efficient to least efficient) until the Pumped Flow has been met or all available pumps have been used.
2. When the Pumped Flow is exceeded, the least efficient pump type is interpolated to determine Pump Power and Pumps Used so that the exact target Pumped Flow is met.
3. If the Pumped Flow cannot be met by all available pumps, the run will abort with an error message that the specified Pumped Flow cannot be met.

4. Pump Energy is calculated as the product of Pump Power and the length of the timestep.

If Pump Power is input by the user, a similar approach is taken to the case where Pumped Flow is input. Entire types of pumps are added until the given Pump Power has been met. When the target Pump Power has been exceeded, linear interpolation is done on the least efficient pump type so that the pump power is met exactly. If all the pump types cannot meet the target Pump Power, the run will abort with an error message that the specified Pump Power cannot be met.

Finally, the Power Coefficient is calculated as Pumped Power divided by Inflow.

21.1.9 Pump and Release Accommodation

This method is used to specify whether pumping and release occur in the same timestep, or whether for a given timestep, the reservoir is either pumping, or releasing.

21.1.9.1 Pump or Release Only

This is the default method. When selected, the reservoir may not pump and release on the same timestep. “Startup” on page -893 for the dispatch methods available when this user method is selected.

21.1.9.2 Pump and Release

This method is used when the user wishes to model pumping and releasing on the same timestep. This is mainly used for post optimization simulation where Pumped Flow and Outflow are both set as input as a result of the optimization run. Even though both Pumped Flow and Outflow are allowed to be inputted, one of the values has to be a zero. If both pumping and releasing are taking place, the user must specify the Pumped Energy and Energy resulting from releasing. “Startup” on page -893 for the dispatch methods available when this user method is selected.

21.1.10 Spill

The Spill methods (except the Monthly Spill which is described in Section 21.1.10.2) calculate the Spill from Reservoirs based on several possible physical combinations of controlled and uncontrolled spillways.

The **Regulated Spill** and **Bypass** slots are regulated (i.e. controlled) spill structures. Values in these two slots can be specified by the user via inputs or rules. Each slot accommodates spill up to the maximum amount as specified by its rating table (**Regulated Spill Table** and **Bypass Table**). **Unregulated Spill** is an uncontrolled spill. Therefore, it is always a computed output based on the average **Pool Elevation** of the reservoir as specified in the **Unregulated Spill Table**. Thus, the user selects a **Spill** method based on the combination of structures (Regulated Spill, Bypass, and/or Unregulated Spill) that exist on the reservoir and the level of granularity desired.

The total **Spill** slot is the sum of the individual spills from each structure. Spills are calculated twice for each timestep. The first time a **Spill** method is called from a dispatch method, it checks for user inputs, calculates any **Unregulated Spill**, and sets the spill to zero for **Regulated Spill** and **Bypass** structures where there is no user-specified value. The total **Spill** is then calculated and returned to the dispatch method. The dispatch method determines (**Turbine**) **Release** by subtracting the **Spill** from **Outflow**, and executes the user-specified power calculation method. (On the power reservoirs, the slot is called **Turbine Release**, on the storage reservoir, the slot is called **Release**. In this description, we use the term (**Turbine**) **Release**.) If the (**Turbine**) **Release** cannot be met in the power calculation method, a second call is made to the spill calculation method. The excess flow is then distributed among the **Regulated Spill** and/or **Bypass** structures which have available capacity. If both **Regulated Spill** and **Bypass** are available, excess spill is typically first discharged through the **Regulated Spill** (except when the "**Bypass, Regulated and Unregulated**" method is selected; for this method the **Bypass** gets spill first).

The optional DRIFT flag is available on the **Regulated Spill** and **Bypass** slots. When the DRIFT flag is set for several sequential timesteps, the method models varying flow through a set spillway gate in response to fluctuations in **Pool Elevation**. The first timestep prior to initializing drift is used to determine a gate index called **Regulated** (or **Bypass**) **Drift Index**. This index is interpolated from the 3-dimensional **Regulated** (or **Bypass**) **Spill Index Table**, which relates **Pool Elevation** to **Spill** for several gate indices. In the subsequent timesteps where the DRIFT flag is set, the same index is used to find the spill value at the current average **Pool Elevation**. The gate index is maintained throughout the selected time period. At each timestep, a new value of spill is calculated for the structure based on the current **Pool Elevation**. Specifying DRIFT is considered an input, and may affect over determination of spill parameters.

21.1.10.1 None

None is used if **Spill** should not be modeled. In this method, the **Spill** slot on the reservoir is set equal to zero. All releases must be through the (**Turbine**) **Release**.

Slots Specific to this Method

☛ None

This method sets the **Spill** to zero and performs no further calculations. If the method is being called for the second time in a timestep because there is excess outflow that won't fit through (**Turbine**) **Release**, an error will be posted which states, "No Spillways Available." In this case, either decrease the **Outflow** or select a different spill method.

21.1.10.2 Monthly Spill

The Monthly Spill method is only appropriate for use in long timestep models where Reservoir fluctuations over the timestep cannot be accurately determined. It is important to note that there is no physical (head dependent) basis to the spill in this method. In this method, there are three components to the spill: unregulated, regulated and bypass. Both Regulated Spill and Bypass are considered controlled releases. A Maximum Controlled Release must be specified by the user.

$$Maximum\ Controlled\ Release = max\ Turbine\ Release + max\ Regulated\ Spill + Bypass$$

Any additional Outflow is immediately categorized as Unregulated Spill.

$$Unregulated\ Spill = Max(0.0, Outflow - Maximum\ Controlled\ Release)$$

Bypass may be specified as a user input. If not input, **RiverWare**™ sets this slot to zero. Regulated Spill is always computed by **RiverWare**™. It is set to zero unless the Reservoir cannot release the Outflow through the Release and Bypass. When this occurs, the additional portion of the Outflow is released through the Regulated Spill.

Slots Specific to this Method

☛ Bypass

Type: Series Slot
Units: FLOW
Description: flow through the Bypass spillway
Information:
I/O: Optional; may be input by the user or set to zero by **RiverWare**™.
Links: Usually not linked

☛ Maximum Controlled Release

Type: Table Slot
Units: FLOW
Description: the maximum amount of Turbine Flow, Regulated Spill, and Bypass
Information: 1X1 table slot
I/O: Required input
Links: Not linkable

☛ Regulated Spill

Type: Series Slot
Units: FLOW
Description: excess Outflow not released through the turbine(s)
Information:
I/O: Output only
Links: Usually not linked

☛ Unregulated Spill

Type: Series Slot
Units: FLOW
Description: Outflow in excess of the Maximum Controlled Release
Information:
I/O: Output only
Links: Not linkable

Initially, the Monthly Spill method is called before Release has been calculated. If the Outflow is greater than the Maximum Controlled Release, Unregulated Spill is set as:

$$Unregulated\ Spill = Outflow - Maximum\ Controlled\ Release$$

If the Outflow is less than the Maximum Controlled Release, Unregulated Spill is set to zero. In both cases, all of the following evaluations are also made:

$$Regulated\ Spill = 0.0$$

$$Bypass = user\ input\ or\ 0.0$$

$$Spill = Unregulated\ Spill + Regulated\ Spill + Bypass$$

After Release is calculated, the Monthly Spill method may be called a second time. The method is called a second time if the Release cannot accommodate the remaining portion of the Outflow:

$$Outflow > Unregulated\ Spill + Release + Bypass$$

Remember that Unregulated Spill was calculated before the Release was calculated.

If this occurs, Regulated Spill and Spill are reevaluated as follows:

$$Regulated\ Spill = Outflow - Unregulated\ Spill - Release - Bypass$$

$$Spill = Unregulated\ Spill + Regulated\ Spill + Bypass$$

21.1.10.3 Unregulated

The Unregulated spill method models a single uncontrolled spillway called **Unregulated Spill**. The **Unregulated Spill** is a function of the average reservoir **Pool Elevation**. Because it is uncontrolled, it takes precedence (i.e. water goes through it first) over other types of outflow (i.e. **Release** or **Turbine Release**) in the reservoir. When this method is chosen the user category Unregulated Spill Type , Section 21.1.11, will appear.

The user may not specify (input or via rules) any spill slots with this method.

Slots Specific to this Method

☛ Unregulated Spill

Type: Series Slot
Units: FLOW
Description: spill corresponding to the average Pool Elevation over the timestep
Information:
I/O: Output only
Links: Not linkable

☛ Unregulated Spill Capacity Fraction

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Unregulated Spill structure that is available.
Information: If not input or set by a rule, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 50 ft of a 1000 ft long crest is blocked, the Unregulated Spill Capacity Fraction would be input to 0.95.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Unregulated Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW

Description: Pool Elevation versus corresponding Unregulated Spill values
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required input
Links: Not linkable

When the Unregulated spill method is called for the first time from the Dispatch Method, Unregulated Spill is calculated and Spill is set equal to Unregulated Spill.

The steps for computing Unregulated Spill are given below:

1. A temporary variable called "initHW" is created to represent the Pool Elevation at the beginning of the timestep. Likewise, "endHW" is created to represent the Pool Elevation at the end of the timestep (If the Pool Elevation at the end of the timestep is not known, endHW is set equal to the Pool Elevation at the beginning of the timestep.)

2. The Unregulated Spillway Crest is set equal to the Pool Elevation that corresponds to a Spill of zero (from the Unregulated Spill Table).

3. If both initHW and endHW are less than or equal to the Unregulated Spillway Crest, Unregulated Spill is set equal to zero.

4. If both initHW and endHW are greater than or equal the Unregulated Spillway Crest, the average Pool Elevation is used to determine the Unregulated Spill from the Unregulated Spill Table.

$$\text{Unregulated Spill} = \text{Value from table} \times \text{Unregulated Spill Capacity Fraction}$$

5. If either initHW or endHW is greater than the Unregulated Spillway Crest and the other is lower than the crest, the following evaluations and computations are performed:

$$\text{maxHW} = \text{the greater of initHW and endHW}$$

$$\text{minHW} = \text{the lesser of initHW and endHW}$$

$$\text{avgHW} = \frac{\text{maxHW} + \text{Unregulated Spillway Crest}}{2}$$

$$\text{spill fraction} = \frac{\text{maxHW} - \text{Unregulated Spillway Crest}}{\text{maxHW} - \text{minHW}}$$

That is, spill fraction corresponds to the fraction of the timestep during which spill occurs.

A temporary variable called “temp spill” is obtained from the linear interpolation of the Unregulated Spill Table using avgHW. Unregulated Spill is then calculated as:

$$\text{Unregulated Spill} = \text{spill fraction} \times \text{temp spill} \times \text{Unregulated Spill Capacity Fraction}$$

The Unregulated Spill is then limited to be less than or equal to the maxUnregulatedSpill if one has been calculated. See below.

The Unregulated spill method will be called a second time (after the (**Turbine**) **Release** has been calculated) only if the sum of (**Turbine**) **Release** and **Spill** are less than **Outflow**. When this is the case, an error which reads, “Outflow greater than spillway capacities and Release” is posted because the excess **Spill** cannot be incorporated.

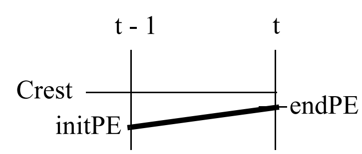
Computing the Maximum Possible Unregulated Spill

If an unregulated spill method is called from the solveMB_givenInflowRelease dispatch method, getMinSpillGivenInflowRelease or solveTurbineRelGivenEnergyInflow RPL predefined functions, there is an upper limit on the unregulated spill. The upper limit prevents the unregulated spill from dropping the reservoir below the crest. The following algorithm is performed at the start of the following methods to compute this maximum unregulated spill (maxUnregulatedSpill):

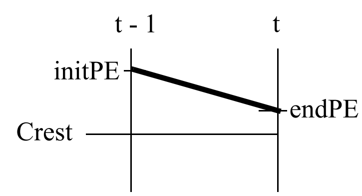
- solveMB_givenInflowRelease dispatch method (Storage and Level Power reservoirs)
- solveMB_givenInflowOutflow dispatch method (Storage and Level Power reservoirs)
- getMinSpillGivenInflowRelease RPL predefined function
- solveTurbineRelGivenEnergyInflow RPL predefined function

1. Given previous Storage and Pool Elevation (initPE), all known inflows (Inflow, Hydrologic Inflow, Precipitation, Return Flow, etc) and known outflows (Release, Regulated Spill, Bypass, Evaporation, Diversion, etc) are used to compute the storage and pool elevation (endPE) that would occur with no additional unregulated spill.

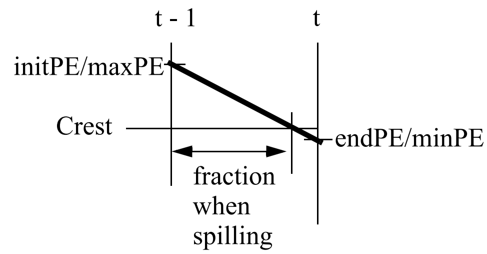
2. If both initPE and endPE are less than or equal to the Unregulated Spillway Crest, maxUnregulatedSpill is set to zero and the computation exits. There is no way there could be unregulated spill. See figure to the right.



3. If both initPE and endPE are greater than or equal to the Unregulated Spillway Crest, the maxUnregulatedSpill is computed as the flow that would draw the reservoir down to exactly reach the crest at the end of the timestep. This computation solves the reservoir mass balance and includes all source and sink terms. All water above the crest could be spilled. See figure to the right.



4. If either initPE or endPE is greater than the Unregulated Spillway Crest and the other is lower than the crest (because of existing Diversions, Evap, etc), the following evaluations and computations are performed: (See figure to the right)



$$\text{maxPE} = \text{the greater of initPE and endPE}$$

$$\text{minPE} = \text{the lesser of initPE and endPE}$$

$$\text{spill fraction} = \frac{\text{maxPE} - \text{Unregulated Spillway Crest}}{\text{maxPE} - \text{minPE}}$$

That is, spill fraction corresponds to the fraction of the timestep during which spill occurs.

5. The storage at the crest, crestStorage, is computed from the Elevation Volume Table.

6. The storage, maxStorage, that corresponds to the maxPE is found on the Elevation Volume Table.

7. The maxUnregulatedSpill (limited to be greater than or equal to zero) is then computed as:

$$\text{maxUnregulatedSpill} = \frac{\text{maxStorage} - \text{crestStorage}}{\text{TimestepLength}} \times \text{spillFraction}$$

The maxUnregulatedSpill is applied as a final limit on the Unregulated Spill. Remember, the Unregulated Spill is computed as described in the start of this method using the Unregulated Spill Table and a similar spill fraction approach; it may already be less than the maxUnregulatedSpill.

21.1.10.4 Regulated

The Regulated spill method models **Spill** using one controlled spillway called **Regulated Spill**. Because the spill is controlled, the spill may be any value between zero and the maximum possible regulated spill for that pool elevation. The user may specify (input or via rules) either:

- No slots
- **Spill** or
- **Regulated Spill**

If either is specified and there is excess flow which cannot be met by the (**Turbine**) **Release**, a **RiverWare**™ error will be flagged and the simulation halted.

Slots Specific to this Method

☛ Regulated Spill

Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare**™. If Regulated Spill is input by the user and the value is less than the required spill, a **RiverWare**™ error is flagged and the simulation is halted.
Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☛ Regulated Spill Capacity Fraction

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Regulated Spill structure that is available.
Information: If not input or set by a rule, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Regulated Spill Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index
Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Regulated Spill Index Table.
Links: Not linkable

☛ Regulated Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding Maximum Regulated Spill values
Information:
I/O: Required input
Links: Not linkable

☛ REGULATED SPILL INDEX TABLE

Type: Table Slot
Units: NOUNITS VS. LENGTH VS. FLOW
Description: Gate Index vs. Pool Elevation vs. Regulated Spill
Information: Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column 3. The table shown below is an example of the proper way to formulate the Regulated Spill Index Table.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130
4	550	180
4	600	230

I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.

Links: Not linkable

Note: Regulated Spill and Spill are both outputs if neither is specified by the user. Only one of these slots, however, can be specified on a given timestep. When this is the case, the other slot will be output.

The first step in the Regulated spill method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 21.1.12.2. If not, the minRegSpill is zero. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

Release is then checked to see if it has been calculated. If Release is not known, it means that the method is being called for the first time for the particular timestep and the following steps are taken:

1. If both the Spill and Regulated Spill are input (remember setting a DRIFT flag is considered an input), a **RiverWare**™ error is flagged and the run is aborted.

2. If Spill is input by the user, and it is greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted. Otherwise, Regulated Spill is set equal to Spill.

3. If Regulated Spill is input/rules by the user and the DRIFT flag is set, a function is called to perform the drift calculations. A description of the DRIFT function is given at the end of this method description. If Regulated Spill is input and greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted. Otherwise, Spill is set equal to the Regulated Spill.

4. If neither Regulated Spill nor Spill are input, they are both set equal to the minRegSpill.

After Release has been calculated, the Regulated spill function may be called a second time if the sum of Release and Spill is less than the Outflow.

The following calculations and evaluations are performed if the function is called for the second time:

1. If either Spill or Regulated Spill are input, a **RiverWare**™ error is posted stating that **RiverWare**™ is unable to allocate the excess flow and the run is aborted.

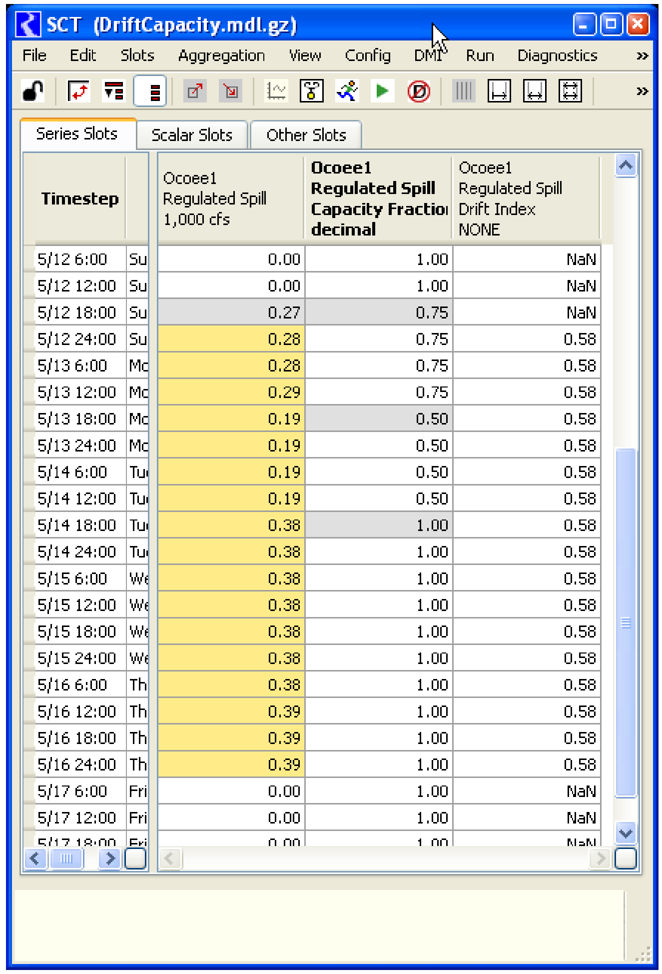
2. Regulated Spill is set equal to the Outflow minus the Release. If Regulated Spill is greater than the maxRegSpill, a **RiverWare**™ error is flagged informing the user that Outflow is greater than the spillway capacities and Release and the run is aborted.

3. Spill is set equal to Regulated Spill.

Drift Calculations:

The drift function is used to calculate Regulated Spill at a specific timestep if it is flagged DRIFT. If the current timestep's Regulated Drift Index is not known, but the previous timestep's Regulated Drift Index is known, the current Regulated Drift Index is set equal to the previous timestep's Regulated Drift Index.

The Drift tables assume that the full spill works is available. Therefore, if there is a Capacity Fraction that is less than 1.0, the Drift calculation (for both regulated and bypass) must be modified.



With the Drift flag is set, if there is a valid Capacity Fraction[t-1] that is not equal to 1.0, then the Capacity Fraction[t] is set to the previous value, but not overwriting inputs or rule values. This causes the Capacity Fraction to remain throughout the drift operation unless it is changed via a new user input. The screenshot to the right shows a sample run. The Capacity fraction is set to 0.75 on 5/12 18:00 and that value remains until a new value is set via user input on 5/13 18:00. Although this set of inputs may not make physical or operations sense, is shows how the algorithm would perform given the inputs shown.

The current Regulated Drift Index is then used in conjunction with the average Pool Elevation over the current timestep and the Regulated Spill Index Table to obtain the current timestep's Regulated Spill.

If it is the first DRIFT timestep, a Gate Index must be calculated. This is done by using an average of the Pool Elevation at t-2 (when available) and at t-1, the previous regulated spill, the greatest and least spill values, and the Regulated Spill Index Table. If there is a non-zero capacity fraction, the value used for previous regulated spill is adjusted as follows:

$$\text{Regulated Spill}[t-1]_{Adj} = \frac{\text{Regulated Spill}[t-1] - \text{Closed Gate Overflow}[t-1]}{\text{Regulated Spill Capacity Fraction}[t-1]}$$

This computes the spill that would have occurred if all the gates were available. If the previous regulated spill is less than the smallest possible Regulated Spill or greater than the largest possible Regulated Spill (according to the Regulated Spill Index Table), a **RiverWare**™ error is flagged and the run is aborted.

The Regulated Drift Index slot is then set for the current timestep. Finally, the Regulated Spill can be determined from the three dimensional interpolation of the Regulated Spill Index Table using the average Pool Elevation over the timestep and the Regulated Spill Index.

At the end of the algorithm, the computed regulated spill (assuming all the gates are available) is multiplied by the Capacity Fraction [t] to determine spill that will occur with the given capacity fraction. This Regulated Spill is then set on the slot.

21.1.10.5 Regulated and Unregulated

The Regulated and Unregulated method models **Spill** through one controlled, **Regulated Spill**, and one uncontrolled spillway, **Unregulated Spill**. First, the **Unregulated Spill** can not be specified (input or via rules) or a **RiverWare**™ error will abort the run. The **Unregulated Spill** is a function of the average reservoir **Pool Elevation** and takes precedence (i.e. water goes through it first) over other types of outflow (i.e. **Release** or **Turbine Release** or **Regulated Spill**) in the reservoir. When this method is chosen the user category Unregulated Spill Type , Section 21.1.11, will appear.

Second, the user may specify (input or via rules) either

- No slots
- **Spill** or
- **Regulated Spill**

If one is specified and there is excess flow which cannot be met by the (**Turbine**) **Release**, an error will be flagged and the simulation halted.

SLOTS SPECIFIC TO THIS METHOD

☛ REGULATED SPILL

Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare**™. If regulated spill is set by the user and the value is greater than the required spill, an error is flagged and the simulation is halted.
Links: Usually not linked. But, it can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☛ REGULATED SPILL CAPACITY FRACTION

Type: Series Slot
Units: DECIMAL

Description: The fraction of the Regulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Regulated Spill Drift Index
Type: Series Slot
Units: NONE
Description: gate setting index
Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Regulated Spill Index Table.
Links: Not linkable

☛ Regulated Spill Index Table
Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Regulated Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.
Links: Not linkable

Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column 3.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130

Gate Index	Pool Elevation	Spill
4	550	180
4	600	230

Regulated Spill Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding Max Regulated Spill values
Information:
I/O: Required input
Links: Not linkable

Unregulated Spill

Type: Series Slot
Units: FLOW
Description: spill corresponding to the average Pool Elevation over the timestep
Information:
I/O: Output only
Links: Not linkable

Unregulated Spill Capacity Fraction

Type: Series Slot
Units: Decimal
Description: The fraction of the Unregulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 50 ft of a 1000 ft long crest is blocked, the Unregulated Spill Capacity Fraction would be input to 0.95.
I/O: Input, set by a rule, or output
Links: Not linkable

Unregulated Spill Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding Unregulated Spill values
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required input
Links: Not linkable

Note: Regulated Spill and Spill are both output slots if neither is input by the user. Only one of these slots, however, can be input for any timestep. When this is the case, the other slot will be an output slot.

The first step in the regulatedPlusUnregSpillCalc method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 21.1.12.2. If not, the minRegSpill is 0.0. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \\ \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

The Unregulated Spill is then calculated through the steps described in the “Unregulated”. If the Unregulated Spill is input by the user, a **RiverWare**™ error is flagged and the run is aborted.

Release is then checked to see if it has been calculated. If Release is not known, it means that the method is being called for the first time for the particular timestep and the following steps are taken:

1. If both the Spill and Regulated Spill are input (remember setting a DRIFT flag is considered an input), a **RiverWare**™ error is flagged and the run is aborted.

2. If Spill is input and it is greater than the sum of the Unregulated Spill and the maxRegSpill, a **RiverWare**™ error is flagged which states that the requested Spill cannot be met. If the Spill is input and less than the Unregulated Spill plus minRegSpill, a **RiverWare**™ error is flagged. Otherwise, Regulated Spill is calculated as Spill minus Unregulated Spill.

3. If Regulated Spill is input by the user and the DRIFT flag is set, a function is called to perform the drift calculations. A description of the DRIFT function is given in the Regulated Spill section. If Regulated Spill is input and greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted. Otherwise, Spill is set equal to the Regulated Spill plus Unregulated Spill.

4. If neither Regulated Spill nor Spill are input, Regulated Spill is set equal to the minRegSpill and Spill is set equal to Unregulated Spill plus minRegSpill.

After the Release has been calculated, the Regulated and Unregulated function may be called a second time if the sum of Release and Spill is less than the Outflow.

The following calculations are performed if the function is called for the second time:

1. If either Spill or Unregulated Spill are input, an error is posted because there are no free spill variables and **RiverWare**™ is unable to allocate the excess Outflow.
2. The Regulated Spill is calculated using the following equation:

$$\text{Regulated Spill} = \text{Outflow} - \text{Turbine Release} - \text{Unregulated Spill}$$

3. If Regulated Spill is greater than the maxRegSpill, a **RiverWare**™ error is posted stating that the Outflow is greater than the spillway capacities and Release and the run is aborted.
4. Spill is calculated as Regulated Spill plus Unregulated Spill.

21.1.10.6 Regulated and Bypass

The Regulated and Bypass method models spill through two regulated spillways called **Regulated Spill** and **Bypass**. The user may specify (input or via rules):

- No slots
- **Spill**
- **Spill** and **Bypass**
- **Spill** and **Regulated Spill**
- **Bypass**
- **Regulated Spill**, or
- **Bypass** and **Regulated Spill**

If all three slots are specified, an error will be issued. Also, if **Spill** is specified and there is excess flow that cannot be met by (**Turbine**) **Release**, a **RiverWare**™ error will be flagged and the simulation halted.

The order in which water goes through the various outflow structures depends on what is known. Input/Rules values take precedence, followed by (**Turbine**) **Release**, followed by **Regulated Spill**, and finally by **Bypass**. For example, on a timestep where there is zero (**Turbine**) **Release** and no spill slots are specified, outflows will first go through **Regulated Spill** and any flow greater than max regulated spill will go through **Bypass**.

Slots Specific to this Method

☛ Bypass

Type: Series Slot
Units: FLOW
Description: flow through the Bypass spillway
Information:

I/O: Optional; may be input by the user or determined by **RiverWare™**. If Bypass is set by the user and the value is greater than the required spill, a **RiverWare™** error is flagged and the simulation is halted.

Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Bypass behaves the same as if it were input.

Bypass Capacity Fraction

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Bypass structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 2 of 8 gates are unavailable, the Bypass Capacity Fraction would be input to 0.75.
I/O: Input, set by a rule, or output
Links: Not linkable

Bypass Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index for the Bypass spillway
Information: If the user has set the DRIFT flag on the Bypass slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Bypass Index Table.
Links: Not linkable

Bypass Index Table

Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Bypass Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Bypass spill slot, this data table must be provided.
Links: Not linkable

Bypass Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding maximum bypass spill values
Information:

I/O: Required input

Links: Not linkable

☛ Regulated Spill

Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare**™.

Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☛ Regulated Spill Capacity Fraction

Type: Series Slot
Units: Decimal
Description: The fraction of the Regulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.
I/O: Input, set by a rule, or output

Links: Not linkable

☛ Regulated Spill Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index for the Regulated Spill
Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Regulated Spill Index Table.

Links: Not linkable

☛ Regulated Spill Index Table

Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Regulated Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.

Links: Not linkable

Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column 3.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130
4	550	180
4	600	230

☛ Regulated Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding maximum regulated spill values
Information:
I/O: Required input
Links: Not linkable

Note: Spill, Regulated Spill, and Bypass can all be outputs if they are not specified by the user. The user may specify Spill, or either Regulated Spill or Bypass, or both Regulated Spill and Bypass. The slots which are not specified will be output slots.

The first step in the Regulated Spill method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 21.1.12.2. If not, the minRegSpill is 0.0. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

The next step is to obtain the maximum Bypass, maxBypass, by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Bypass Table. Then the Bypass Capacity Fraction is applied as follows:

$$\text{maxBypass} = \text{value from Bypass Table} \times \text{Bypass Capacity Fraction}$$

1. If Spill, Bypass, and Regulated Spill are all input/rules, a **RiverWare**™ Error is posted and the simulation run is aborted.

2. If Spill and Regulated Spill are input/rules by the user, the following steps are performed (Remember: If the DRIFT flag is set on the Regulated Spill slot, Regulated Spill is considered an input.):
   - Drift calculations are performed if the DRIFT flag is set on the Regulated Spill slot. See Regulated Spill for a description of the drift calculations.
   - If Regulated Spill is greater than either Spill or maxRegSpill a **RiverWare**™ error is flagged and the run is aborted.
   - If Regulated Spill is less than minRegSpill, a **RiverWare**™ error is flagged and the run is aborted.
   - Bypass is calculated as Spill minus Regulated Spill.
   - If Bypass is greater than the maxBypass, an error is flagged.

3. If Spill and Bypass are input, the following steps are taken (Remember: If the DRIFT flag is set on the Bypass slot, Bypass is considered an input.):
   - Drift calculations are performed if the DRIFT flag is set on the Bypass slot. The DRIFT calculations are performed in a similar manner to the drift calculations for Regulated Spill which are explained in the Regulated Spill method description.
   - If Bypass is greater than either Spill or the maxBypass, a **RiverWare**™ error is flagged and the simulation is aborted.
   - Regulated Spill is calculated as Spill minus Bypass.
   - If Regulated Spill is either greater than the maximum regulated spill or less than the minimum regulated spill, an error is flagged and the run is aborted.

4. If Spill is input but neither Regulated Spill nor Bypass are input or flagged as DRIFT, the following steps are taken:
   - Regulated Spill is set as the lesser value of either the Spill or the maxRegspill. The Regulated Spill also cannot be less than the minRegSpill.

- If Regulated Spill is less than Spill, Bypass is calculated as Spill minus Regulated Spill.
- If Bypass is greater than the maxBypass, a **RiverWare**™ error is flagged and the run is aborted.
- If Regulated Spill is equal to Spill, Bypass is set equal zero.

5. If Spill is not input, and both Bypass and Regulated Spill are input, the following steps are taken:
   - The drift calculations are performed for both Regulated Spill and Bypass if the DRIFT flags have been set. A description of the DRIFT calculations is contained in the Regulated Spill method.
   - Regulated Spill and Bypass are checked against the maxRegSpill and maxBypass, respectively. If either Regulated Spill is greater, a **RiverWare**™ error is flagged and the run is aborted.
   - Spill is calculated as Regulated Spill plus Bypass.
6. If the DRIFT flag is set on the Regulated Spill, the drift calculations are performed (as described in the Regulated Spill method) to calculate Regulated Spill. The calculated Regulated Spill value is then checked against spillway capacities. If only Regulated Spill is input, the value is checked against the spillway capacity. Spill is set equal to Regulated Spill and Bypass is set to zero if the Regulated Spill is less than or equal to the maxRegSpill. If the Regulated Spill is greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted and the simulation run is aborted.
7. If the DRIFT flag is set on the Bypass, the drift calculations are performed (as described in the Regulated Spill method) to calculate Bypass. The calculated Bypass value is then checked against spillway capacities. If only Bypass is input, the input value is checked against the spillway capacity. Spill is calculated as Bypass plus Regulated Spill. Regulated Spill is set to the minimum regulated spill, if the Bypass is less than the maximum bypass. If the Bypass is greater than the maxBypass, a **RiverWare**™ error is posted and the simulation run is aborted.
8. If no slots are input, Spill is set equal minRegSpill. Bypass is set to zero and Regulated Spill is set to minRegSpill.

After Release has been calculated, the Regulated and Bypass function may be called a second time if the sum of the Release and Spill is less than the Outflow.

The following calculations are performed if the function is called for the second time:

1. If either Spill is input, or both Regulated Spill and Bypass are input, a **RiverWare**™ error is flagged and the simulation run is aborted because there are no free spill variables.

2. If only Regulated Spill is input or flagged as DRIFT, Bypass is recalculated using the following formula:

$$\text{Bypass} = \text{Outflow} - \text{Regulated Spill} - \text{Turbine Release}$$

Bypass is then checked against its spillway capacity and a **RiverWare**™ error is flagged and the simulation is aborted the spillway capacity is exceeded. Spill is calculated as Bypass plus Regulated Spill if the Bypass is less than or equal to maximum allowable bypass.

3. If only Bypass is input or flagged as DRIFT, Regulated Spill is recalculated using the following formula:

$$\text{Regulated Spill} = \text{Outflow} - \text{Bypass} - \text{Turbine Release}$$

The Regulated Spill is then checked against its spillway capacity and minRegSpill. A **RiverWare**™ error is posted and the simulation run is aborted if the spillway capacity is exceeded or is less than minimum. Spill is calculated as the sum of Bypass and Regulated Spill.

4. If neither Bypass nor Regulated Spill is input, the following steps are performed:
   - A local variable, excess, is calculated as Outflow minus Turbine Release minus minRegSpill.
   - Regulated Spill is set equal to the lesser value of excess or maxRegSpill but must be greater than minRegSpill.
   - If Regulated Spill is less than the excess, Bypass is calculated as excess minus Regulated Spill.
   - Bypass is checked against its spillway capacity. If Bypass is greater than the maxBypass, a **RiverWare**™ error is posted and the simulation run is aborted.
   - Spill is calculated as the Bypass plus the Regulated Spill.
   - If Regulated Spill is equal to the Excess, Bypass is set equal to zero and Spill is set equal to Regulated Spill.

21.1.10.7 Regulated, Bypass and Unregulated

This method models spill through two controlled spillways called **Bypass** and **Regulated Spill** and one uncontrolled spillway called **Unregulated Spill**. The user may not specify (input or via rules) the **Unregulated Spill**. This value is always output and is a function of the average reservoir Pool Elevation. The user may specify (input or rules):

- No slots
- **Spill**

- **Spill** and **Bypass**
- **Spill** and **Regulated Spill**
- **Bypass**
- **Regulated Spill**, or
- **Bypass** and **Regulated Spill**

If **Spill**, **Regulated Spill**, and **Bypass** are specified, an error will be issued. Also, if **Spill** is specified and there is excess flow that cannot be met by (**Turbine**) **Release**, a **RiverWare**™ error will be flagged and the simulation halted.

The order in which water goes through the various outflow structures depends on what is known. **Unregulated Spill** takes precedence, followed by input/rules values, followed by (**Turbine**) **Release**, followed by **Regulated Spill**, and finally by **Bypass**. For example, on a timestep where there is zero (**Turbine**) **Release** and no spill slots are specified, outflows will first go through **Unregulated Spill** (computed based on pool elevation), then **Regulated Spill** up to capacity and any excess flows will go through **Bypass**.

Slots Specific to this Method

☛ Bypass

Type: Series Slot
Units: FLOW
Description: flow through the Bypass spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare**™. If Bypass is set by the user and the value is greater than the required spill, a **RiverWare**™ error is flagged and the simulation is halted.
Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Bypass behaves the same as if it were input.

☛ Bypass Capacity Fraction

Type: Series Slot
Units: Decimal
Description: The fraction of the Bypass structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 2 of 8 gates are unavailable, the Bypass Capacity Fraction would be input to 0.75.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Bypass Drift Index

Type: Series Slot

Units: NONE
Description: gate setting index for the Bypass spillway
Information: If the user has set the DRIFT flag on the Bypass slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Bypass Index Table.
Links: Not linkable

☛ Bypass Index Table

Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Bypass Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Bypass spill slot, this data table must be provided.
Links: Not linkable

☛ Bypass Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding maximum bypass spill values
Information:
I/O: Required input
Links: Not linkable

☛ Regulated Spill

Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare**™.
Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☛ Regulated Spill Capacity Fraction

Type: Series Slot
Units: Decimal
Description: The fraction of the Regulated Spill structure that is available.

Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Regulated Spill Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index for the Regulated Spill
Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if this slot is not set by the user, the gate index is calculated from the Regulated Spill Index Table.
Links: Not linkable

☛ Regulated Spill Index Table

Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Regulated Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.
Links: Not linkable

Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column 3.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130
4	550	180
4	600	230

☛ Regulated Spill Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding maximum regulated spill values
Information:
I/O: Required input
Links: Not linkable

☛ Unregulated Spill

Type: Series Slot
Units: FLOW
Description: spill corresponding to the average Pool Elevation over the timestep
Information:
I/O: Output only
Links: Not linkable

☛ Unregulated Spill Capacity Fraction

Type: Series Slot
Units: Decimal
Description: The fraction of the Unregulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 50 ft of a 1000 ft long crest is blocked, the Unregulated Spill Capacity Fraction would be input to 0.95.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Unregulated Spill Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding unregulated spill values
Information:
I/O: Required input
Links: Not linkable

Note: Spill, Regulated Spill, and Bypass may be output slots if they are not specified as input by the user. The user may specify either Spill or Regulated Spill or Bypass, or both Unregulated Spill and Bypass as input. The slots which are not set as input will be output slots.

The first step in the Regulated Spill method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 21.1.12.2. If not, the minRegSpill is 0.0. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

The next step is to obtain the maximum Bypass, maxBypass by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Bypass Table. Then the Bypass Capacity Fraction is applied as follows:

$$\text{maxBypass} = \text{value from Bypass Table} \times \text{Bypass Capacity Fraction}$$

The Unregulated Spill is then calculated through the steps described in the “Unregulated”. If the Unregulated Spill is input by the user, a **RiverWare**™ error is flagged and the run is aborted.

If Spill, Bypass, and Regulated Spill are input, Spill is overdetermined and a **RiverWare**™ error is flagged and the simulation run is aborted.

If Release has not been calculated, the method is executing for the first time in the current timestep and the following steps are taken:

1. If Spill is input by the user and Unregulated Spill is greater than the Spill, a **RiverWare**™ error is flagged and the simulation run is aborted.
2. If Spill and Regulated Spill are input the following steps are taken (Remember: If the DRIFT flag is set on the Regulated Spill slot, Regulated Spill is considered an input):
    - Drift calculations are performed if the DRIFT flag is set on the Regulated Spill slot. See Regulated Spill for a description of the drift calculations.
    - If Regulated Spill is greater than either Spill or maxRegSpill, a **RiverWare**™ error is flagged and the run is aborted.
    - If Regulated Spill is less than minRegSpill, a **RiverWare**™ error is flagged and the run is aborted.
    - Bypass is calculated as Spill minus Regulated Spill minus Unregulated Spill.
    - If Bypass is either greater than the maxBypass or less than zero, an error is flagged.
3. If Spill and Bypass are input, the following steps are taken (Remember: If the DRIFT flag is set on the Bypass slot, Bypass is considered an input.):

- Drift calculations are performed if the DRIFT flag is set on the Bypass slot. The DRIFT calculations are performed in a similar manner to the drift calculations for Regulated Spill, which are explained in the Regulated Spill method description.
- If Bypass is greater than either Spill or the maxBypass, a **RiverWare**™ error is flagged and the run is aborted.
- Regulated Spill is calculated as Spill minus Bypass minus Unregulated Spill.
- If Regulated Spill is either greater than the maxRegSpill or less than the minRegSpill, an error is flagged and the run is aborted.

4. If Spill is input by the user and neither Bypass nor Regulated Spill are input, the following steps are taken:

- Regulated Spill is set as the lesser value of either the Spill minus Unregulated Spill or the maxRegSpill. The Regulated Spill also cannot be less than the minRegSpill.
- If Regulated Spill is less than Spill minus Unregulated Spill, Bypass is set equal to Spill minus Regulated Spill minus Unregulated Spill.
- If Bypass is greater than the maxBypass, a **RiverWare**™ error is flagged and the run is aborted.
- If Regulated Spill is equal to Spill minus Unregulated Spill, Bypass is set equal to zero.

5. If Spill is not input, but both Bypass and Regulated Spill are input, the following steps are taken:

- The drift calculations are performed for both Regulated Spill and Bypass if the DRIFT flags have been set. A description of the DRIFT calculations is contained in the Regulated Spill method.
- Regulated Spill and Bypass are checked against the maxRegSpill and maxBypass, respectively. If either Regulated Spill is greater, a **RiverWare**™ error is flagged and the run is aborted. If Regulated Spill is less than the minRegSpill, an error is issued.
- Spill is calculated as the sum of Regulated Spill, Bypass, and Unregulated Spill.

6. If the DRIFT flag is set on the Regulated Spill, the drift calculations are performed (as described in the Regulated Spill method) to calculate Regulated Spill. The calculated Regulated Spill Value is then checked against spillway capacities. If only Regulated Spill is input, the value is checked against the spillway capacity. Spill is calculated as Regulated Spill plus Unregulated Spill and Bypass is set to zero, if the Regulated Spill is less than the maxRegSpill. If the Regulated Spill is greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted and the simulation run is aborted.

7. If the DRIFT flag is set on the Bypass, the drift calculations are performed (as discussed in the Regulated Spill method) to calculated the Bypass. The calculated Bypass value is then checked against spillway capacities. If only Bypass is input, the input value is checked against the spillway capacity. Spill is calculated as Bypass plus Unregulated Spill and Regulated Spill is set to the minRegSpill, if the Bypass is less than the maxBypass. If the Bypass is greater than the maxBypass, a **RiverWare**™ error is posted and the simulation run is aborted.

8. If no slots are input, Spill is set equal to Unregulated Spill plus minRegSpill. Bypass is set to zero and Regulated Spill is set to minRegSpill.

After Release has been calculated, the Regulated, Bypass and Unregulated function may be called a second time if the sum of the Release and Spill is less than the Outflow.

The following calculations are performed if the function is called for the second time:

1. If either Spill is input or both Regulated Spill and Bypass are input, a **RiverWare**™ error is flagged and the simulation run is aborted because there are no free spill variables.

2. If only Regulated Spill is input or flagged as DRIFT, Bypass is recalculated using the following formula:

$$\text{Bypass} = \text{Outflow} - \text{Regulated Spill} - \text{Unregulated Spill} - \text{Turbine Release}$$

The Bypass is then checked against its spillway capacity. A **RiverWare**™ error is posted and the simulation run is aborted if the spillway capacity is exceeded. Spill is calculated as the sum of Bypass, Unregulated Spill, and Regulated Spill if the Bypass is less than or equal to the maxBypass.

3. If only Bypass is input, Regulated Spill is recalculating the following formula:

$$\text{Regulated Spill} = \text{Outflow} - \text{Bypass} - \text{Unregulated Spill} - \text{Turbine Release}$$

The Regulated Spill is then checked against its spillway capacity and minRegSpill. A **RiverWare**™ error is posted and the simulation run is aborted if the spillway capacity is exceeded or is less than minimum. Spill is calculated as the sum of Bypass, Regulated Spill, and Unregulated Spill if the Regulated Spill is less than or equal to the maximum allowable regulated spill.

4. If neither Regulated Spill nor Bypass are input, the following steps are performed:
   - A local variable, excess, is calculated as Outflow minus Unregulated Spill minus Turbine Release minus minimum regulated spill.
   - Regulated Spill is set equal to the lesser value of excess or maxRegSpill but must be greater than minRegSpill.

- If Regulated Spill is less than excess, Bypass is calculated as excess minus Regulated Spill.
- Bypass is checked against its spillway capacity. If Bypass is greater than the maxBypass, a **RiverWare™** error is posted and the simulation run is aborted.
- Spill is set equal to the sum of Bypass, Regulated Spill, and Unregulated Spill.
- If Regulated Spill is equal to Excess, Bypass is set equal to zero and Spill is set equal to the sum of Regulated Spill and Unregulated Spill.

21.1.10.8 Bypass, Regulated and Unregulated

Note: This user method is the similar to the **Regulated, Bypass and Unregulated** method but switches the order of the **Bypass** and **Regulated Spill** outlet works. This method is preferable in institutional cases where the term "**Bypass**" is favored over the term "**Regulated Spill**". Other than the order reversal, the functionality is similar to the Regulated, Bypass and Unregulated method.

This method models spill through two controlled spillways called **Bypass** and **Regulated Spill** and one uncontrolled spillway called **Unregulated Spill**. The user may not specify (input or via rules) the **Unregulated Spill**. This value is always output and is a function of the average reservoir **Pool Elevation**. The user may specify (input or rules):

- No slots
- **Spill**
- **Spill** and **Bypass**
- **Spill** and **Regulated Spill**
- **Bypass**
- **Regulated Spill**, or
- **Bypass** and **Regulated Spill**

If **Spill**, **Regulated Spill**, and **Bypass** are specified, an error will be issued. Also, if **Spill** is specified and there is excess flow that cannot be met by (**Turbine**) **Release**, a **RiverWare™** error will be flagged and the simulation halted.

The order in which water will go through the various outflow structures depends on what is known. **Unregulated Spill** takes precedence, followed by input/rules values, followed by (**Turbine**) **Release**, followed by **Bypass**, and finally by **Regulated Spill**. For example, on a timestep where there is zero (**Turbine**) **Release** and no spill slots are specified, outflows will first go through **Unregulated Spill** (required based on pool elevation), then **Bypass** up to capacity and any excess flows will go through **Regulated Spill**.

Please see the **regPlusBypassPlusUnregSpill** method for a description of the slots particular to this method and the algorithm of this method. The algorithm for this method is only different in that **Bypass** takes precedence over **Regulated Spill**.

21.1.11 Unregulated Spill Type

This category is only visible when a method using **Unregulated Spill** is chosen. The three Unregulated Spill Types are Bare Crest Only, Two Unregulated Flow, and Three Unregulated Flows.

21.1.11.1 Bare Crest Only

The Bare Crest Only method is the default method in the Unregulated Spill Type Category. The method assumes an unobstructed spillway where the flow over the spillway is a function of the **Unregulated Spill Table**. There are no slots specifically associated with this method.

21.1.11.2 Two Unregulated Flows

When the **Two Unregulated Flows** method is selected, flow over the spillway is a function of the **Unreg Flow 2 Spill Table.** If the pool elevation meets or exceeds the **Unreg Flow 2 Failure Elevation**, flow becomes a function of the **Unregulated Spill Table.**

Note: This method originally was called **Flashboards** but was renamed to be more general. (Flashboards are wooden boards installed in the unregulated spillway so that the reservoir may store more water than what the spillways themselves would allow.)

☛ Unreg Flow 2 Avail and Failure Time

Type: Agg Series Slot
Units: FRACTION
Description: Availability of Unreg Flow 2 Spill Table, fraction of timestep when Unreg Flow 2 Spill Table is in use.
Information:
I/O: Optional: Availability may be input by user or set by a rule. Failure time is output only
Links: Not linkable

☛ Unreg Flow 2 Failure Elevation

Type: Table Slot
Units: LENGTH
Description: Pool Elevation at which Unreg Flow 2 Spill Table is no longer used.
Information:
I/O: Required Input
Links: Not linkable

☛ Unreg Flow 2 Spill Table

Type: Table Slot
Units: Length vs. Flow
Description: Pool Elevation vs. corresponding unregulated flow 2 spill values.
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required Input
Links: Not linkable

Method Details

To determine the spill tables to use during a timestep based on availability of Unreg Flow 2, the following three possibilities are checked:

1. Availability is input or set by a rule: no change from Unreg Flow 2 to Unregulated Spill over timestep, calculate spill based on availability.
2. Availability is 0: no change from Unreg Flow 2 to Unregulated Spill, calculate spill based on Unregulated Spill Table.
3. Availability is greater than 0, check if failure from Unreg Flow 2 to Unregulated Spill occurs during the timestep.

During the third case, the spill over the timestep is calculated using the respective table and multiplied by the availability to find the total spill over the timestep. The total spill is used to predict the pool elevation at the end of the timestep. The calculated pool elevation is compared to the Unreg Flow 2 Failure Elevation. If a failure from Unreg Flow 2 to Unregulated Spill is found to occur any time during the timestep, the failure time is recorded and the ending pool elevation is re-calculated to account for the change in spill due to the change from Unreg Flow 2 to Unregulated Spill during the timestep.

Note: If a failure from Unreg Flow 2 to Unregulated Spill occurs during a dispatch, the first time of this failure is used for the remainder of the dispatch.

The time of failure during the timestep is used to determine what portion of the timestep needs interpolation from each of the two spill tables.

21.1.11.3 Three Unregulated Flows

When the **Three Unregulated Flows** is selected, flow over the spillway is a function of the **Unreg Flow 3 Spill Table.** If the pool elevation meets or exceeds the **Unreg Flow 3 Failure Elevation**, flow becomes a function of the **Unreg Flow 2 Spill Table.** If the pool elevation meets or exceeds the **Unreg Flow 2 Failure Elevation**, flow becomes a function of the **Unreg Flow Spill Table.** To summarize, Unreg Flow 3 fails first, then Unreg Flow 2 fails next as the pool rises. Therefore, the Unreg Flow 2 Failure Elevation should be higher than the Unreg Flow 3 Failure Elevation.

Note: This method was originally called the **Flashboards and Superboards** method but was renamed to be more general. Flashboards and superboards are wooden boards installed in the unregulated spillway so that the reservoir may store more water than what the spillways themselves would allow. The superboards can only be installed if the flashboards are in place.

☛ Unreg Flow Avail and Failure Time

Type: Agg Series Slot
Units: FRACTION
Description: Availability and failure time of Unreg flow 2 and 3 Spill tables.
Information:
I/O: Optional: Availability may be input by user or set by a rule. Failure time is output only
Links: Not linkable

☛ Unreg Flow 3 Failure Elevation

Type: Table Slot
Units: LENGTH
Description: Pool Elevation at which Unreg Flow 3 Spill Table is no longer used.
Information: The value in this slot should be lower than the value in the Unreg Flow 2 Failure Elevation.
I/O: Required Input
Links: Not linkable

☛ Unreg Flow 3 Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding unregulated flow 3 spill values.
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required Input
Links: Not linkable

☛ Unreg Flow 2 Failure Elevation

Type: Table Slot
Units: LENGTH
Description: Pool Elevation at which Unreg Flow 2 Spill Table is no longer used.
Information: The value in this slot should be higher than the value in the Unreg Flow 2 Failure Elevation.

I/O: Required Input

Links: Not linkable

☛ Unreg Flow 2 Spill Table

Type: Table Slot
Units: Length vs. Flow
Description: Pool Elevation vs. corresponding unregulated flow 2 spill values.
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required Input

Links: Not linkable

Method Details

To determine the spill tables to use during a timestep based on availability of Unreg Spill 2 and Unreg Spill 3, the following four possibilities are checked:

1. Both availabilities are input or set by a rule: no change in unregulated spill type over timestep, calculate spill based on availability.

2. One availability is input or set by a rule: Error, both or none must be input.

3. Availabilities are 0. No change in unregulated spill type over timestep. Calculate spill based on Unregulated Spill Table.

4. At least one availability is greater than 0, check for failure in unregulated spill type.

During the fourth case the spill over the timestep is calculated using the respective table and multiplied by the availability to find the total spill over the timestep. The total spill is used to project the pool elevation at the end of the timestep. The calculated pool elevation is compared to the failure elevations of Unreg Flow 2 or Unreg Flow 3. If failure in unregulated spill type is found to occur any time during the timestep, this failure time is recorded and the ending pool elevation is recalculated to account for the change in spill due to the change in unregulated spill type during the timestep.

Note: If a change in unregulated spill type occurs during a dispatch, the first time of this change is used for the remainder of the dispatch.

The time at which change in unregulated spill type occurred during the timestep is used to determine what portion of the timestep needs interpolation from each of the three spill tables.

21.1.12 Regulated Spill Overflow

The category, **Regulated Spill Overflow,** is added if one of the following "regulated" spill methods is selected:

- **Regulated Spill**
- **Regulated and Unregulated**
- **Regulated and Bypass**
- **Regulated, Bypass and Unregulated**
- **Bypass, Regulated and Unregulated**

21.1.12.1 None

This is the default, no-action method.

21.1.12.2 Closed Gate Overflow

This method models the uncontrolled flow over a closed regulated spill gate. This functionality uses the **Regulated Spill Capacity Fraction** to compute the default amount of spillway that is overtopped.

This functionality only applies to **Regulated Spill**, not **Bypass**.

THE FOLLOWING SLOTS WILL BE ADDED:

☛ **CLOSED GATE OVERFLOW**

Type: Series Slot
Units: FLOW
Description: Uncontrolled portion of the Regulated spill that overtops the gates.
Information: This value is computed by the regulated spill method as the value found on the **Closed Gate Overflow Table** multiplied by the **Closed Gate Overflow Capacity Fraction**
I/O: Output only
Links: Not linkable

☛ **CLOSED GATE OVERFLOW TABLE**

Type: Table Slot
Units: LENGTH VS FLOW
Description: Pool Elevation vs unregulated flow.
Information: This table is used to specify the rating curve for uncontrolled flow over the closed Regulated Spill gates. The values should be input to the table as though every Regulated Spill gate is closed. The tables would start with zero flow at or just below the top of the closed gates.
I/O: Input Only

Links: Not available

☛ Closed Gate Overflow Capacity Fraction

Type: Series Slot
Units: FRACTION
Description: The fraction of the closed gate overflow that is available.
Information: The value must be between 0.0 and 1.0, inclusive. If not input or set by a rule, it defaults to (1 - **Regulated Spill Capacity Fraction**). Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be set to 0.875 and the Closed Gate Overflow Capacity Fraction would default to 0.125.
I/O: Input, set by a rule, or output
Links: Not linkable

Method Details

If not input or set by a rule, **Closed Gate Overflow Capacity Fraction** defaults to (1 - **Regulated Spill Capacity Fraction**). This default indicates that the overflow only happens over gates that are closed. Otherwise, the user can specify the **Closed Gate Overflow Capacity Fraction** slot to say how much of the overflow structure is available.

When the reservoir is below the top of the gates, there is no **Closed Gate Overflow**. But once the reservoir is above the top of one or more closed gates, there is **Closed Gate Overflow**. The computation of this overflow is similar to the unregulated spill computation:

1. A temporary variable called "initHW" is created to represent the Pool Elevation at the beginning of the timestep. Likewise, "endHW" is created to represent the Pool Elevation at the end of the timestep (If the Pool Elevation at the end of the timestep is not known, endHW is set equal to the Pool Elevation at the beginning of the timestep.)

2. The "Closed Gate Overflow Crest" is found from the Closed Gate Overflow Table. It is the Pool Elevation that corresponds to an overflow of zero.

3. If both initHW and endHW are less than or equal to the Closed Gate Overflow Crest, Closed Gate Overflow is set equal to zero.

4. If both initHW and endHW are greater than the Closed Gate Overflow Crest, the average Pool Elevation is used to determine the Closed Gate Overflow from the Closed Gate Overflow Table.

$$\text{Closed Gate Overflow} = \text{Value from table} \times \text{Closed Gate Overflow Capacity Fraction}$$

5. If either initHW or endHW is greater than the Closed Gate Overflow Crest and the other is lower than the crest, the following evaluations and computations are performed:

$$\text{maxHW} = \text{the greater of initHW and endHW}$$

$$\text{minHW} = \text{the lesser of initHW and endHW}$$

$$\text{avgHW} = \frac{\text{maxHW} + \text{Closed Gate Overflow Crest}}{2}$$

$$\text{overflow fraction} = \frac{\text{maxHW} - \text{Closed Gate Overflow Crest}}{\text{maxHW} - \text{minHW}}$$

where:

maxHW = the maximum value of Pool Elevation over the timestep.

minHW = the minimum value of Pool Elevation over the timestep.

avgHW = the average Pool Elevation causing overflow over the timestep.

overflow fraction = corresponds to the fraction of the timestep during which overflow occurs.

A temporary variable called “temp overflow” is obtained from the linear interpolation of the Closed Gate Overflow Table using avgHW. Closed Gate Overflow is then calculated as:

$$\text{Closed Gate Overflow} = \text{overflow fraction} \times \text{temp overflow} \times \text{Closed Gate Overflow Capacity Fraction}$$

When allocating spills to various structures, the **Closed Gate Overflow** must occur at the same time as unregulated spills (i.e. before regulated or bypass). Then any remaining outflow can go through the regulated and or bypass spill structures. Therefore, the minimum regulated spill is computed as follows:

$$\text{Min Regulated Spill} = \\ \text{value from Closed Gate Overflow Table} \times \text{Closed Gate Overflow Capacity Fraction}$$

Also, the **Closed Gate Overflow** is set equal to the Min Regulated Spill.

The functionality assumes that water is either flowing through the gate or over topping it, but not both. The method assumes that the Gate Overflow Table is fixed, that is the elevations in the table do not change. Thus if you had a gate stuck with 1/2 ft open at the bottom and there was still water going over the top, the table (which assumes the gate is closed) would be an incorrect rating.

21.1.13 Input Outflow Adjustment

This method category is only available if a method is selected in the Spill Calculation category. Its purpose is to adjust input Outflow values if they violate a physical constraint.

21.1.13.1 None

This is the default method. It performs no calculations and there are no slots associated with it. The Outflow values will not be adjusted if this method is selected.

21.1.13.2 Reduce Input Outflow

This method is used to reduce the input Outflow value whenever it exceeds the maximum reservoir outflow (due to outlet works capacity).

Slots Specific to this Method

☛ **Requested Outflow**

Type: Series Slot
Units: FLOW
Description: The Outflow value before being adjusted
Information: This slot is available so that the user can see when an Outflow value is adjusted. The value in this slot is the outflow value before being adjusted. A value exists in this slot only if the Outflow value is adjusted.
I/O: Output only
Links: Not linkable

If the Outflow slot value is greater than the maximum reservoir outflow, this method saves the Outflow value in the Requested Outflow slot. Then, the Maximum Capacity flag is set on the Outflow slot. The reservoir is then forced to re-dispatch with the Outflow set to Max Capacity (instead of the original, input value). When the reservoir solves the second time, it computes the maximum reservoir outflow and sets this value on the Outflow slot. The Maximum Capacity flag remains on the Outflow slot for the timestep in question (and will be saved with the model file).

21.1.13.3 Allow Excess Specified Outflows

This method allows input Outflows that exceed the maximum reservoir outflow (due to outlet capacity). Because the excess is above the maximum possible for the (Turbine) Release and Spill slots, it will not be classified as either. Instead, the excess is stored on a separate series slot for reporting or tracking.

Slots Associated With this Method:

☛ Outflow Exceeding Max

Type: Series Slot
Units: Flow
Description: The portion of the input Outflow that exceeds the sum of the Spill and (Turbine) Release.
Information: This slot tracks the amount that does not fit through the Release and Spill structures.
I/O: Output Only
Links: Not Linkable

Method Details:

Toward the end of each dispatch method, if Outflow is greater than the sum of Spill and (Turbine) Release, the Spill method is executed again to redistribute the Outflows to the appropriate spill structures. Within the Spill method, if there is still **no** room for the specified Outflow, the selected method in the **Input Outflow Adjustment** category is executed. When the **Allow Excess Specified Outflows** method is selected, it does the following:

If the Outflow does not have an input flag (I or Z), then the method exits and issues an error that there are excess outflows.

If the Outflow is input (I or Z flag), the method computes the difference between the specified Outflow and the maximum Outflow (i.e. Turbine Release + Spill). This excess outflow is then set on the **Outflow Exceeding Max** slot.

$$\text{Outflow Exceeding Max}[\,] = \text{Outflow}[\,] - ((\text{Turbine})\ \text{Release}[\,] + \text{Spill}[\,])$$

The method then exits successfully and returns to the Spill method and then the dispatch method. The dispatch method sets the spill and mass balance slots.

21.1.14

21.1.15 Future Value

The methods in this category are used to determine the future value of the energy that would have been generated by the water that was lost through the spillway.

21.1.15.1 None

None is the default method for the Future Value category. No calculations are performed by this method. There are no slots specifically associated with this method.

21.1.15.2 Cumulative Storage Value Table

Slots Specific to this Method

☛ **Marginal Storage Value Table**

Type: Table
Units: Volume vs. $per Energy
Description: Storage versus marginal value per unit energy
Information: This table should be increasing in storage, and usually decreasing in marginal value.
I/O: Required input
Links: Not linkable

☛ **Spill Cost**

Type: Series Slot
Units: $
Description: Future cost of energy lost due to spilled water
Information:
I/O: Output only
Links: May be linked to the Spill Cost slot on the Thermal Object.

☛ **Future Value of Used Energy**

Type: Series Slot
Units: $
Description: Future value of energy used in the current timestep
Information:
I/O: Output only
Links: May be linked to Future Value of Used Energy Slot on the Thermal Object.

☛ **Anticipated Storage**

Type: Series

Units: VOLUME
Description: The combination of Storage in the reservoir at the given timestep plus any flow (converted volume) that is in transit to the reservoir
Information: This slot represents the storage including any lagged flows that are already in a linked upstream reach, which will reach the reservoir at a later timestep. If there are no lagged reaches between this reservoir and the next upstream reservoir, Anticipated Storage will equal Storage. It is this storage value that will be used to calculate Cumulative Storage Value.
I/O: Output only
Links: May be linked

CUMULATIVE STORAGE VALUE

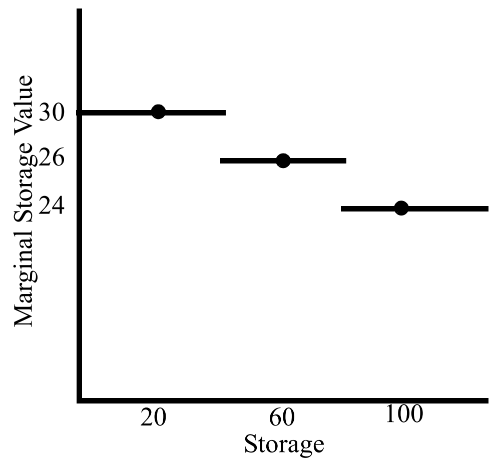
Type: Series Slot
Units: $
Description: Represents the future energy value of the current Anticipated Storage
Information:
I/O: Output only
Links: May be linked to the Total Cumulative Storage Value Slot on the Thermal Object

CUMULATIVE STORAGE VALUE TABLE

Type: Table
Units: VOLUME VS. $
Description: Anticipated Storage and cumulative value used to calculate the Cumulative Storage Value as a function of Anticipated Storage
Information: This table should be increasing in storage and usually increasing in cumulative storage value.
I/O: Required Input either by the user or automatically generated by **RiverWare**™ if the Cumulative Storage Value Table Automation method is selected.
Links: Not linkable

This method uses the Marginal Storage Value Table and the calculated Spill and Turbine Release to compute the Spill Cost and Future Value of Used Energy. These are the only calculations performed by this method.

The correct marginal value is found from the current storage in the reservoir. If the current storage is than the midpoint between the first and second storage table values, the first marginal value is used. The second marginal value is used for a current storage above that midpoint to the midpoint between the second and third storage table values. The last marginal value is used for any current storage above the midpoint between the second-to-last and the last storage table value. An example is shown in Table 16 on page 832



Storage	Marginal Value
20	30
60	26
100	24

Table: 16 Marginal Value Table

Assume that the current storage is 39. Therefore, this method would use 30 as the marginal value for use in the next computation. Assume that the current storage is 41. Therefore, this method would use 26 as the marginal value for use in the next computation.

Use of a table in this fashion is unique to this method.

Spill Cost is computed by the following equation:

$$\text{Spill Cost} = \text{Spill} \times \text{Marginal Storage Value} \times \text{Timestep Length}$$

Future Value of Used Energy is computed by the following equation:

$$\text{Future Value of Used Energy} = \text{Turbine Release} \times \text{Marginal Storage Value} \times \text{Timestep Length}$$

The Cumulative Storage Value computation begins by first calculating Anticipated Storage. This is the sum of the reservoir Storage plus any flow already in transit to the reservoir in an upstream lagged reach. For example, assume a reservoir’s Inflow slot is linked to a reach with a 3-hour lag time. In an hourly run, the reservoir’s Anticipated Storage would be calculated as:

$$\text{Reservoir.Anticipated Storage} = \text{Reservoir.Storage} + (\text{Reach.Inflow(-2)} + \text{Reach.Inflow(-1)} + \text{Reach.Inflow}) \times TimestepLength$$

If there are no lagged reaches between the reservoir and the next upstream reservoir, then Anticipated Storage will simply equal Storage.

The Cumulative Storage Value is then computed by interpolating from the Cumulative Storage Value Table using the calculated Anticipated Storage value.

21.1.16 Cumulative Storage Value Table Automation

This category allows the **RiverWare**™ simulation to automate the creation of the Cumulative Storage Value Table. This category is only visible if Cumulative Storage Value Table is selected in the Future Value category.

21.1.16.1 None

If this method is selected, no automation will be performed and the user must enter the data into the Cumulative Storage Value Table.

21.1.16.2 Marginal Value to Table

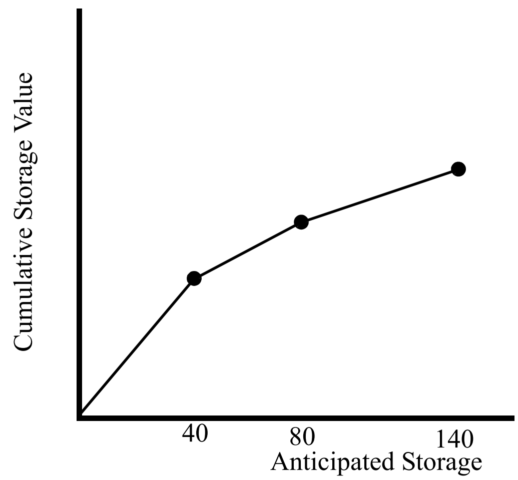
If this method is selected, the Marginal Storage Value table will be used as the source for the generation of the Cumulative Storage Value Table. This is the only calculation associated with this method. There are no slots associated specifically with this method.

This method uses information from the simulation slot Marginal Storage Value Table to generate a cumulative storage value table. The cumulative storage value can be thought of as the summation of the marginal storage values from a storage of 0 to the current storage. Therefore, the automation method finds the same midpoint values used by the simulation Future Value method, and uses those points in the table.

If only one value exists in the Marginal Storage Value Table, then only two entries will exist in the Cumulative Storage Value Table. The two entries will be 0, and midway between the value in the Marginal Storage Value Table, and the maximum value set on the Storage slot. If more than one value exists, three or more points will result. An example is shown below of a Marginal Value Table and the resulting Cumulative Storage Value Table with a graph of the Cumulative Storage Value data.

Storage	Marginal Value
20	30
60	26
100	24

Table: 17 Marginal Value Table



Anticipated Storage	Cumulative Value
40	1200
80	2240
140	3680

Table: 18 Cumulative Storage Value Table

21.1.17 Ramping

This category allows you to model the cost of turbine ramping, the cost of changing the turbine release from one timestep to the next.

21.1.17.1 None

This is the default method for this category. No new slots are instantiated, and no calculations are performed.

21.1.17.2 Track Ramping

In this method the user assigns a unit cost to turbine ramping, a cost per unit change in flow from one timestep to the next. The method then calculates the change in turbine release and the associated ramping cost. The unit cost is the same for ramping up and ramping down.

SLOTS SPECIFIC TO THIS METHOD

☛ RAMPING COST

Type: Series
Units: VALUE
Description: The total cost of turbine ramping for the timestep.
Information: This is the Unit Ramping Cost multiplied by the change in turbine release from one timestep to the net.
I/O: Output only
Links: May be linked

☛ TURBINE DECREASE

Type: Series
Units: FLOW
Description: The difference between Turbine Release at the previous timestep and the current timestep when Turbine Release decreases from the previous timestep
Information: If Turbine Release at the current timestep is greater than at the previous timestep, the value is zero.
I/O: Output only
Links: May be linked

☛ TURBINE INCREASE

Type: Series
Units: FLOW
Description: The difference between Turbine Release at the current timestep and the previous timestep when Turbine Release increases from the previous timestep

Information: If Turbine Release at the current timestep is less than at the previous timestep, the value is zero.
I/O: Output only
Links: May be linked

Unit Ramping Cost

Type: Table 1x1
Units: Value per Flow
Description: The cost per unit change in Turbine Release from one timestep to the next
Information: There is a single value for Unit Ramping Cost (i.e. the same unit cost for ramping up and ramping down)
I/O: Optional input; if not input or negative, defaults to zero
Links: Not linkable

The method first checks for a value in the Unit Ramping Cost table slot. If there is no input value, or if the value is negative, it sets the value to zero. Then it checks if Turbine Release is valid for the current timestep and the previous timestep. If Turbine Release at the current timestep is greater than or equal to the previous timestep then the following values are set:

$$\text{Turbine Increase} = \text{Turbine Release} - \text{Turbine Release}(-1)$$

$$\text{Turbine Decrease} = 0$$

$$\text{Ramping Cost} = \text{Unit Ramping Cost} \times \text{Turbine Increase}$$

Otherwise:

$$\text{Turbine Decrease} = \text{Turbine Release}(-1) - \text{Turbine Release}$$

$$\text{Turbine Increase} = 0$$

$$\text{Ramping Cost} = \text{Unit Ramping Cost} \times \text{Turbine Decrease}$$

If Turbine Release is not valid at both the current and previous timesteps, then Turbine Increase, Turbine Decrease and Ramping Cost will all display NaN.

21.1.18 Hydrologic Inflow

The Hydrologic Inflow category allows **RiverWare**™ to accommodate inflows to a Reservoir that are not part of the main channel and/or are not gauged. The user methods in this category may be used to initialize the Hydrologic Inflow slot if it is required by the user. If the Hydrologic Inflow slot has been initialized, it is figured into the mass balance when the object dispatches.

21.1.18.1 None

None is the default method for the Hydrologic Inflow category. No calculations are performed by this method. There are no slots specifically associated with this method. If this method is selected, the Hydrologic Inflow slot is not initialized so it is no included in the mass balance.

21.1.18.2 Input Hydrologic Inflow

The Input Hydrologic Inflow method should be used when the user wishes either to input the values of Hydrologic Inflow or have the values default to zero. **RiverWare**™ will not overwrite any user input values.

Slots Specific to this Method

☛ **Hydrologic Inflow**

Type: Series
Units: FLOW
Description: flow into the reservoir that is not gauged and/or does not enter through the main channel.
Information:
I/O: Optional; defaults to zero if not input.
Links: Usually input or calculated but could be linked to the Outflow of any object or any other series slot.

☛ **Hydrologic Inflow Adjust**

Type: Series
Units: FLOW
Description: optional adjustment that can be made to the calculated Hydrologic Inflow
Information:
I/O: Optional; set to zero if not input by the user.
Links: Not linkable

☛ Hydrologic Inflow Net

Type: Series
Units: FLOW
Description: sum of hydrologic Inflow and Hydrologic Inflow Adjust
Information:
I/O: Output only
Links: Not linkable

The algorithm used for this method is very simple. If Hydrologic Inflow is not input by the user, it is set equal to zero. Hydrologic Inflow Net is calculated as the sum of Hydrologic Inflow and Hydrologic Inflow Adjust.

21.1.19 Gate Setting

Used to determine the gate setting on the pumps which results in optimum performance.

21.1.19.1 None

This is the default method. It performs no gate setting calculations. There are no specific slots associated with this method.

21.1.19.2 Best Gate Setting Table

Used to calculate the best gate setting for the pumps' intake.

Slots Specific to this Method

☛ Best Gate Setting

Type: Table
Units: LENGTH vs. NONE
Description: head vs. gate setting for best pump performance
Information:
I/O: Required input
Links: Not linkable

☛ Gate Setting

Type: Series Slot
Units: NONE
Description: best gate setting for the pump's intake
Information:
I/O: Output only
Links: Usually not linked

The Gate Setting slot is calculated from interpolation of the Best Gate Setting table using the Operating Head for the current timestep.

21.1.20 Evaporation and Precipitation

The Evaporation and Precipitation category methods are used to calculate the volume of Evaporation from and Precipitation to the surface of a reservoir over the timestep. Precipitation and Evaporation are used in the mass balance equations which are solved in the dispatch methods.

Some of the methods in this category only calculate evaporation.

21.1.20.1 None

The None method is the default method for the Evaporation and Precipitation category. It should be chosen if the user does not want to include Evaporation in the mass balance equation of the Reservoir. There are no slots specifically associated with this method. No calculations are performed by this method.

21.1.20.2 Daily Evaporation

The Daily Evaporation method is used to calculate the daily evaporation volume and the flow rate of the precipitation. The daily evaporation volume is a function of the Evaporation Rate, average Surface Area, and Pan Coefficient.

SLOTS SPECIFIC TO THIS METHOD

☛ **ELEVATION AREA TABLE**
Type: Table Slot
Units: LENGTH VS. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☛ **EVAPORATION**
Type: SeriesSlot
Units: VOLUME
Description: volume of water lost to evaporation during one timestep
Information:
I/O: Output only
Links: Not linkable

☛ **EVAPORATION TABLE**
Type: Table Slot
Units: NOUNITS VS. LENGTH (PER DAY)

Description: day of the year vs. Evaporation Rate
Information: The first of January is 0.
I/O: Required Input
Links: Not linkable

Pan Evaporation Coefficient
Type: Table Slot
Units: NO UNITS
Description: a fractional value between 0 and 1 that represents the portion of potential evaporation which actually occurs
Information:
I/O: Required Input
Links: Not linkable

Precipitation Rate
Type: SeriesSlot
Units: LENGTH VS. TIME
Description: precipitation intensity for the given timestep
Information:
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

Precipitation Volume
Type: SeriesSlot
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

Surface Area
Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area calculated from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

The Surface Area is determined using the Pool Elevation and the Elevation Area Table. The Evaporation Rate is looked up in the Evaporation Table according to the current day of the year. Evaporation is calculated using the following equation:

$$\text{Evaporation} = \text{Evaporation Rate} \times \text{Pan Evaporation Coefficient} \times (\text{Surface Area} + \text{Surface Area}(-1))/2$$

The volume of Precipitation that occurred over the timestep is then calculated with the following equation:

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area}(-1))/2$$

where in the above equations:

Evaporation Rate = the Evaporation Rate corresponding to the current day of the year

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

21.1.20.3 Input Evaporation

The Input Evaporation method should be used when the user wants to input the Evaporation Rate directly. This Evaporation Rate is used to compute the volume of water that evaporated over the timestep. Also, the user can input the evaporation volume directly on the Evaporation slot. In that case, the Evaporation Rate is not used to calculated Evaporation.

Slots Specific to this Method

☛ Elevation Area Table

Type: Table Slot
Units: LENGTH VS. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☛ Evaporation

Type: SeriesSlot
Units: VOLUME
Description: volume of water lost due to evaporation during the timestep
Information:
I/O: Output; optional input overrides calculation
Links: Not linkable

☛ Evaporation Rate

Type: SeriesSlot
Units: LENGTH PER TIME
Description: rate at which water evaporates from the surface
Information:
I/O: Optional input, disaggregated by method as described in the Evap and Precip Rate Specification category, or defaults to 0.0 if not specified by the user.
Links: Not linkable

☛ Precipitation Rate

Type: SeriesSlot
Units: LENGTH PER TIME
Description: precipitation intensity for a given timestep
Information:
I/O: Optional input, disaggregated by method as described in the Evap and Precip Rate Specification category, or defaults to 0.0 if not specified by the user.
Links: Not linkable

☛ Precipitation Volume

Type: SeriesSlot
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☛ Surface Area

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

At the beginning of the run, the chosen method in the **Evap and Precip Rate Specification** category is executed. This category allows you to specify the rates as monthly or periodic slots.

If the user specifies Evaporation directly (via input or rules), the value will be used instead of calculating a value below.

If Evaporation is not specified, the following equation is used to compute the volume of water that evaporated from the Reservoir over the timestep:

$$\text{Evaporation} = \text{Evaporation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)}) / 2$$

The precipitation flow rate over the timestep is calculated as shown in the following equation:

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)}) / 2$$

where in the above equations:

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

21.1.20.4 Monthly Evaporation

In the Monthly Evaporation method, evaporation is calculated linearly from the Evaporation Coefficients entered for each month. This method will not work with a timestep longer than monthly. The total evaporated volume is a function of the average Reservoir Surface Area over the timestep, the Evaporation Coefficient, and the length of the timestep. The following slots are specifically associated with this method.

Slots Specific to this Method

☛ Elevation Area Table

Type: TableSlot
Units: LENGTH VS. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☛ Evaporation

Type: SeriesSlot
Units: VOLUME
Description: volume of water lost to evaporation during one timestep
Information:
I/O: Output only
Links: Not linkable

☛ Evaporation Coefficients

Type: TableSlot
Units: LENGTH PER TIME
Description: rate of evaporation for each month
Information: This slot contains one column of values. The Evaporation Coefficient for each month of the year must be input by the user beginning with the Evaporation Coefficient for January.
I/O: Required input
Links: Not linkable

☛ Precipitation Rate

Type: SeriesSlot
Units: LENGTH PER TIME
Description: precipitation intensity for the given timestep
Information: Value must be input by the user for each timestep.
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

☛ Precipitation Volume

Type: SeriesSlot
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☛ Surface Area

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area calculated from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

The Surface Area of the Reservoir is calculated based on the Elevation Area Table. The Evaporation is then calculated using the following formula:

$$\text{Evaporation} = \text{Evaporation Coefficient} \times (\text{Surface Area} + \text{Surface Area }(-1))/2 \times TimestepLength$$

The volume of Precipitation that occurred over the timestep is then calculated using the following equation:

$$\text{Precipitation} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)})/2 \times TimestepLength$$

where in the above equations:

Evaporation Coefficient = the Evaporation Coefficient for the current month

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

21.1.20.5 Pan and Ice Evaporation

The Pan and Ice Evaporation method is used to calculate the volume of evaporation with one of two methods based on the value of the Pan Ice Switch slot for each timestep. The Pan Ice Switch slot is used as an indicator of whether ice is present on the surface of the reservoir. A value of 1.0 in the Pan Ice Switch slot indicates that there is ice cover on the Reservoir that must be taken into account when Evaporation is calculated. A value of 0.0 or any number other than 1.0 in the Pan Ice Switch slot indicates that there is no ice on the surface of Reservoir. The following slots are those specifically associated with this method.

Slots Specific to this Method

☛ Evaporation

Type: Series
Units: VOLUME
Description: volume of water lost to evaporation during the current timestep
Information:
I/O: Output only
Links: Not linkable

☛ Elevation Area Table

Type: Table
Units: LENGTH vs. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☛ K Factor

Type: Series Slot with Periodic Input
Units: VELOCITY PER TEMPERATURE_F
Description: factor relating average temperature, in degrees Fahrenheit, to evaporation rate

Information: This slot is a series slot, but the data can be input as a periodic relationship.
I/O: Optional but is required Input if the Pan Ice Switch slot is 1.0
Links: Not linkable

☛ Max Air Temperature
Type: Series
Units: TEMPERATURE IN FARENHEIT
Description: maximum air temperature during the timestep
Information:
I/O: Optional; required only if the Pan Ice Switch slot is 1.0
Links: Not linkable

☛ Min Air Temperature
Type: Series
Units: TEMPERATURE IN FARENHEIT
Description: minimum air temperature during the timestep
Information:
I/O: Optional; required if the Pan Ice Switch slot is 1.0
Links: Not linkable

☛ Pan Evaporation
Type: Series
Units: LENGTH PER TIME
Description: evaporation rate from the surface
Information:
I/O: Optional; only required if the Pan Ice Switch is 0.0
Links: Not linkable

☛ Pan Evaporation Coefficient
Type: Table
Units: DECIMAL
Description: weighing factor for pan evaporation rate
Information:
I/O: Optional; required if the Pan Ice Switch slot is 0.0
Links: Not linkable

☛ Pan Ice Switch
Type: Series
Units: No Units

Description: indicator of surface ice coverage for each timestep; **1.0** = ice; any other number or **0.0** = no ice.
Information: This slot is a series slot, but the data can be input as a periodic relationship.
I/O: Required input
Links: Not linkable

☛ Precipitation Rate

Type: Series
Units: LENGTH PER TIME
Description: precipitation intensity for a given timestep
Information:
I/O: Optional; defaults to 0.0 if not specified by the user.
Links: Not linkable

☛ Precipitation Volume

Type: Series
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☛ Surface Area

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

☛ Surface Ice Coverage

Type: SeriesSlot
Units: DECIMAL
Description: fraction of the Surface Area which is covered by ice
Information:
I/O: Optional; only used if the Pan Ice Switch slot is 1.0. Defaults to 0.0 for any timestep not specified by the user.
Links: Not linkable

If the Pan Ice Switch slot is equal to 1.0, ice is present and the following calculation is performed to compute evaporation:

$$\text{Evaporation} = \frac{\text{Max Air Temperature} + \text{Min Air Temperature}}{2} \times \text{K Factor} \times (1 - \text{Surface Ice Coverage}) \times \text{average Surface Area} \times \mathit{Timestep}$$

If the calculated Evaporation is less than zero, the Evaporation is set equal to zero.

The Precipitation is calculated with the following equation if the Pan Ice Switch slot is equal to 1.0:

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times (1 - \text{Surface Ice Coverage}) \times \text{average Surface Area}$$

The volume of precipitation that accumulated over the timestep at the Reservoir (Precipitation Volume) is the product of the precipitation flow rate and the timestep.

If the Pan Ice Switch slot is 0.0 or any number other than 1.0, there is no ice and the following calculation is performed to compute Evaporation:

$$\text{Evaporation} = \text{Pan Evaporation} \times \text{Pan Evaporation Coefficient} \times \text{average Surface Area} \times \text{Timestep}$$

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times \text{average Surface Area}$$

The volume of precipitation that accumulated over the timestep at the Reservoir (Precipitation Volume) is the product of the precipitation flow rate and the timestep.

21.1.20.6 Pan and Ice Evaporation, Current Surface Area

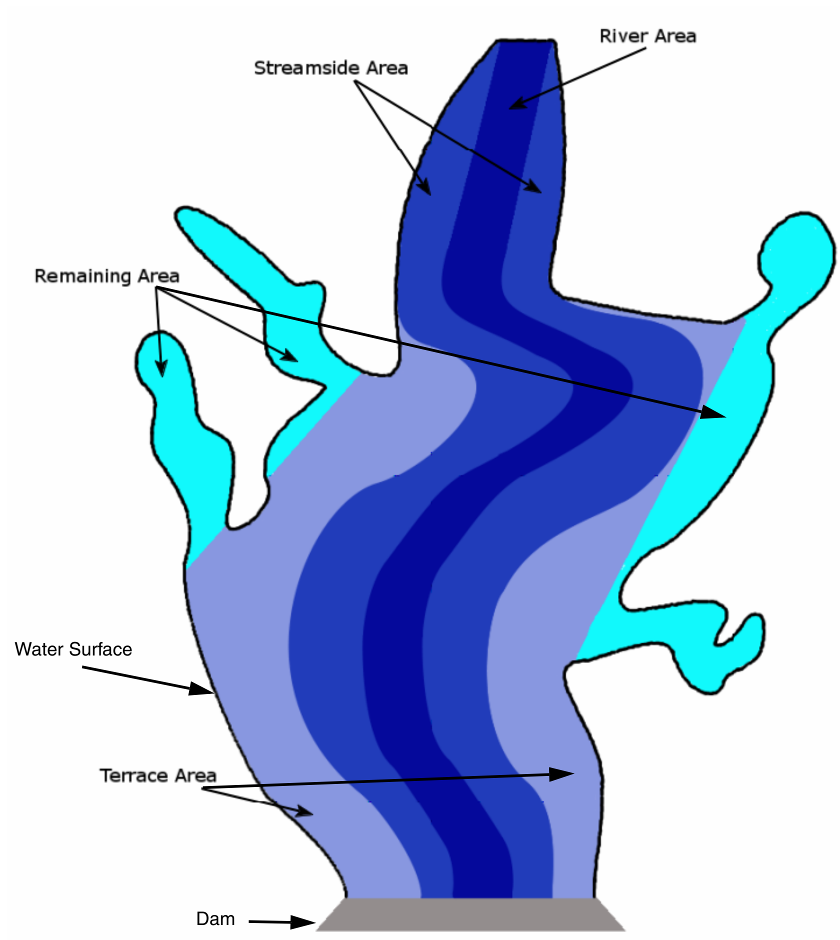
This method is exactly the same as the Pan and Ice Evaporation method. It uses the same slots, has the same required inputs and performs the same calculations. The only difference is that this method uses the instantaneous, end of timestep surface area instead of the average surface area over the timestep.

21.1.20.7 Periodic Net Evaporation

Periodic Net Evaporation computes the gross evaporation from the reservoir and then subtracts out components of evaporation that would have occurred if the reservoir had not been built. This is the net evaporation and is set in the **Evaporation** slot. Each area of the submerged reservoir is separate including:

- River
- Streamside
- Terrace, and any
- Remaining areas

Each can have a separate evaporation coefficient and possibly additional components in its computation like temperature. The area of each region is specified in a separate table relating reservoir pool elevation to each region's area. It is assumed that any precipitation that falls on the Remaining Area would have completely evaporated. The figure shows a sample of the different reservoir areas used in this method.



Slots Specific to this Method

☛ Average Precipitation

Type: Periodic
Units: Velocity (Length per Time)
Description: Slot describing the average precipitation
Information: Typically this would have a yearly period and monthly precipitation values. It is used in the computation of Remaining Evaporation.
I/O: Required Input
Links: Not Linkable

☛ Average Air Temperature

Type: Periodic
Units: TemperatureInFahren
Description: Slot describing the average air temperature

Information: Typically this would have a yearly period and monthly temperature values. It is used in computation of Streamside Evaporation and Terrace Evaporation.
I/O: Required Input
Links: Not Linkable

☛ Elevation Area Table
Type: Table
Units: Length vs Area
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required Input
Links: Not Linkable

☛ Elevation River Area
Type: Table
Units: Length vs Area
Description: Table relating reservoir Pool Elevation to submerged river area.
Information:
I/O: Required Input
Links: Not Linkable

☛ Elevation Streamside Area
Type: Table
Units: Length vs Area
Description: Table relating reservoir Pool Elevation to submerged streamside area.
Information:
I/O: Required Input
Links: Not Linkable

☛ Elevation Terrace Area
Type: Table
Units: Length vs Area
Description: Table relating Pool Elevation to submerged terrace area
Information:
I/O: Required Input
Links: Not Linkable

☛ Evaporation
Type: Series
Units: Volume

Description: Water lost from the reservoir to evaporation. This is the net evaporation and is the value that is included in the reservoir mass balance.
Information: This is the calculated as Gross Evaporation minus Salvage Evaporation
I/O: Output only
Links: Not Linkable

☛ Gross Evaporation

Type: Series
Units: VOLUME
Description: The total evaporation off the reservoir surface. This is the evaporation that is actually occurring from the reservoir.
Information: This is calculated as GrossEvaporationCoeff times SurfaceAreaAvg converted from a flow to volume.
I/O: Output only
Links: Not Linkable

☛ Gross Evaporation Coefficient

Type: Periodic
Units: VELOCITY (LENGTH PER TIME)
Description: A table that describes the gross evaporation coefficient as it varies periodically. This is similar to a pan evaporation coefficient.
Information:
I/O: Required Input
Links: Not Linkable

☛ River Evaporation Coefficient

Type: Periodic
Units: VELOCITY (LENGTH PER TIME)
Description: A table that describes the river evaporation coefficient as it varies periodically. This is similar to a pan evaporation coefficient.
Information:
I/O: Required Input
Links: Not Linkable

☛ Salvage Evaporation

Type: Series
Units: VOLUME
Description: The evaporation that would have occurred if the reservoir were not in place.

$$\begin{aligned} SalvageEvaporation = {} & RiverEvaporation \\ & + StreamsideEvaporation \\ & + TerraceEvaporation \\ & + RemainingEvaporation \end{aligned}$$

Information:
I/O: Output Only
Links: Not Linkable

☛ STREAMSIDE COEFFICIENT
Type: Periodic
Units: VELOCITYPERTEMPERATURE_F (I.E. LENGTH PER TIME PER TEMPERATURE_F)
Description: Periodic table of coefficients
Information: Typically this represents a unit depth per month per degree Fahrenheit (e.g. inches per month per degree Fahrenheit)
I/O: Required Input
Links: Not Linkable

☛ SURFACE AREA
Type: Series
Units: AREA
Description: Reservoir surface area computed from a lookup on the Elevation Area table
Information:
I/O: Output only
Links: Not Linkable

☛ TERRACE COEFFICIENT
Type: Periodic
Units: VELOCITYPERTEMPERATURE_F (I.E. LENGTH PER TIME PER TEMPERATURE_F)
Description: Periodic table of coefficients
Information: Typically this represents a unit depth per month per degree Fahrenheit (e.g. inches per month per degree Fahrenheit)
I/O: Required Input
Links: Not Linkable

The method will be passed in a current estimate of Surface Area and Average Surface Area. The latter is an average of the current estimate and previous timestep's value. In this description, it is called SurfaceAreaAvg. Similarly, StreamsideAreaAvg, RiverAreaAvg, TerraceAreaAvg and RemainingAreaAvg are all averages of the current and previous values.

In the following steps, the FlowToVolume and VolumeToFlow notation indicates that the specified expression will be converted from a flow to a volume (or vice versa) using the

timestep length. This is necessary for the units to work correctly as evaporation [Volume units] is computed as a coefficient [Length/Time units] times an area [L^2 units]. Note that in the following steps, the slots are in bold while intermediate values are not.

The method does the following:

1. Get the value from the periodic **Gross Evaporation Coefficient** slot. If not valid, issue an error.

2. Compute **Gross Evaporation**:

$$GrossEvaporation = FlowToVolume(GrossEvaporationCoefficient \times SurfaceAreaAvg)$$

3. Get the value from the periodic **River Evaporation Coefficient** slot. If not valid, issue an error.

4. Look up the Pool Elevation at t and t-1 on the **Elevation River Area** table to get the RiverArea at t and t-1. Then

$$RiverAreaAvg = \frac{RiverArea[t] + RiverArea[t-1]}{2}$$

5. Compute River Evaporation:

$$RiverEvaporation = FlowToVolume(RiverEvaporationCoefficient \times RiverAreaAvg)$$

This simulates that the river evaporation that would have occurred without the reservoir is a function of area and coefficient.

6. Get the value from the periodic **Streamside Coefficient** slot. If not valid, issue an error.

7. Look up the Pool Elevation at t and t-1 on the **Elevation Streamside Area** table to get the StreamsideArea at t and t-1. Then:

$$StreamsideAreaAvg = \frac{StreamsideArea[t] + StreamsideArea[t-1]}{2}$$

8. Get the value from the periodic **AverageTemperature** slot. If not valid, issue an error.

9. Compute Streamside Evaporation:

$$StreamsideEvaporation = FlowToVolume(StreamsideCoefficient \times StreamsideAreaAvg \times AverageAirTemperature)$$

This simulates that the streamside evaporation that would have occurred without the reservoir is a function of area, coefficient, and average air temperature.

10. Get the value from the periodic **Terrace Coefficient** slot. If not valid, issue an error.

11. Look up the Pool Elevation at t and t-1 on the **Elevation Terrace Area** table to get the Terrace Area at t and t-1. Then:

$$TerraceAreaAvg = \frac{TerraceArea[t] + TerraceArea[t-1]}{2}$$

12. Compute Terrace Evaporation:

$$\begin{aligned} TerraceEvaporation = {} & FlowToVolume(TerraceCoefficient \times TerraceAreaAvg \\ & \times AverageAirTemperature) \end{aligned}$$

This simulates that the terrace evaporation that would have occurred without the reservoir is a function of area, coefficient, and average air temperature.

13. Compute the average Remaining Area as:

$$\begin{aligned} RemainingAreaAvg = {} & SurfaceAreaAvg - RiverAreaAvg \\ & - StreamsideAreaAvg - TerraceAreaAvg \end{aligned}$$

If RemainingAreaAvg is less than zero, an error will be issued as the table data is incorrect.

14. Get the value from the periodic **Average Precipitation**. If not valid, issue an error.

15. Compute RemainingEvaporation:

$$\begin{aligned} & RemainingEvaporation = \\ & FlowToVolume(RemainingArea \times AveragePrecipitation) \end{aligned}$$

This simulates that all of the precipitation on the Remaining Area would have evaporated.

16. Compute **Salvage Evaporation**:

$$\begin{aligned} SalvageEvaporation = {} & RiverEvaporation \\ & + StreamsideEvaporation \\ & + TerraceEvaporation \\ & + RemainingEvaporation \end{aligned}$$

17. Compute **Evaporation** as follows:

$$Evaporation = GrossEvaporation - SalvageEvaporation$$

The **Evaporation** is then a volume that is removed from the reservoir mass balance in the dispatch method.

21.1.20.8 Single Evaporation

In the Single Evaporation method, evaporation is calculated linearly from the Single Evaporation Coefficient entered by the user. The total evaporated volume is a function of the average Reservoir Surface Area over the timestep, the Single Evaporation Coefficient, and the length of the timestep. The following slots are specifically associated with this method.

Slots Specific to this Method

☛ Elevation Area Table

Type: Table Slot
Units: LENGTH VS. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☛ Evaporation

Type: SeriesSlot
Units: VOLUME
Description: volume of water lost to evaporation during one timestep
Information:
I/O: Output only
Links: Not linkable

☛ Single Evap Coeff

Type: Table Slot
Units: LENGTH PER TIME
Description: rate of evaporation
Information: This slot contains a single value that represents the evaporation rate.
I/O: Required input
Links: Not linkable

☛ Precipitation Rate

Type: SeriesSlot

Units: LENGTH PER TIME
Description: precipitation intensity for the given timestep
Information: Value must be input by the user for each timestep.
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

☛ PRECIPITATION VOLUME

Type: SeriesSlot
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☛ SURFACE AREA

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area calculated from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

The Surface Area of the Reservoir is calculated based on the Elevation Area Table. The Evaporation is then calculated using the following formula:

$$\text{Evaporation} = \text{Evaporation Coefficient} \times (\text{Surface Area} + \text{Surface Area (-1)})/2 \times TimestepLength$$

The volume of Precipitation that occurred over the timestep is then calculated using the following equation:

$$\text{Precipitation} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)})/2 \times TimestepLength$$

where in the above equations:

Evaporation Coefficient = SingleEvapCoeff entered by the user

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

21.1.21 Evap and Precip Rate Specification

This category allows you to choose how the evaporation and precipitation rates will be specified. The category is only available when the **Input Evaporation** method in the **Evaporation and Precipitation** category is specified.

21.1.21.1 None

This is the default method; that is, the rates must be input, set by a rule, or they default to 0.0.

21.1.21.2 Monthly Rates

This method allows you to specify the evaporation and precipitation rates as a series of monthly values for the entire run.

☛ EVAPORATION RATE MONTHLY

Type: Series Slot
Units: VELOCITY
Description: The evaporation rate for each month of the run.
Information: You must set the timestep for this series slot to be monthly. Because this slot is monthly, it is most likely different than the run timestep. As a result, if you “synchronize objects”, you must select the toggle in the synchronization control to “Exclude Slots with Different Timestep from Run.” This will prevent changing the timestep of this slot when other slots are synchronized.
I/O: Optional input
Links: Not linkable

☛ PRECIPITATION RATE MONTHLY

Type: Series Slot
Units: VELOCITY
Description: The precipitation rate for each month of the run.
Information: You must set the timestep for this series slot to be monthly. Because this slot is monthly, it is most likely different than the run timestep. As a result, if you “synchronize objects”, you must select the toggle in the synchronization control to “Exclude Slots with Different Timestep from Run.” This will prevent changing the timestep of this slot when other slots are synchronized.
I/O: Optional input
Links: Not linkable

METHOD DETAILS

At the beginning of run, the method disaggregates the **Evaporation Rate Monthly** and **Precipitation Rate Monthly** to the **Evaporation Rate** and **Precipitation Rate** slots,

respectively. If the timestep of the run is monthly, it uses the values directly. If the timestep of the run is less than a month, it **look ups** the month that contains the given timestep and uses that value. No interpolation is performed.

If the run timestep is annual, an error is issued.

If the two slots are not monthly but have inputs, an error is issued.

If there is no value in the monthly slot for a given month, then the rate is set to 0.0.

21.1.21.3 Periodic Rates

This method allows you to specify the evaporation and precipitation rates as a periodic relationship.

☛ Evaporation Rate Periodic

Type: Periodic Slot
Units: Velocity
Description: The evaporation rate as a periodic relationship.
Information: Like other periodic slots, you can choose the period and whether to interpolate or lookup.
I/O: Required Input
Links: Not linkable

☛ Precipitation Rate Periodic

Type: Periodic Slot
Units: Velocity
Description: The precipitation rate as a periodic relationship.
Information: Like other periodic slots, you can choose the period and whether to interpolate or lookup. If you do not wish to model precipitation, you still must enter a zero in this periodic slot.
I/O: Required Input
Links: Not linkable

Method Details

At the beginning of run, the method sets the **Evaporation Rate** and **Precipitation Rate** slots by looking up (or interpolating as configured on the periodic slot) the given timestep in the **Evaporation Rate Periodic** and **Precipitation Rate Periodic** slots, respectively. If accessing the periodic slot fails due to missing values, then an error is issued and the run stops.

21.1.22 Low Flow Releases

This category is only used to add the slots necessary for low flow release calculations. These slots are generally used by a RPL function (called MeetLowFlowRequirement) to compute the low flow releases necessary to meet the low flow requirements on control point objects.

21.1.22.1 None

This method performs no calculations and adds no slots.

21.1.22.2 Enable Low Flow Releases

This method performs no calculations. It simply adds the Low Flow Release slot and Maximum Low Flow Delivery Rate slot.

SLOTS SPECIFIC TO THIS METHOD

☛ **LOW FLOW RELEASE**

Type: Series Slot
Units: FLOW
Description: The portion of the Outflow that is intended to meet a low flow requirement
Information: This slot is normally computed by a RPL function (MeetLowFlowRequirement) that computes the low flow releases necessary to meet the low flow requirements on control point objects.
I/O: Usually set by a rule
Links: Not linkable

☛ **MAXIMUM LOW FLOW DELIVERY RATE**

Type: Periodic Slot
Units: FLOW
Description: The maximum low flow delivery rate for the reservoir
Information: This value is used by the RPL function (MeetLowFlowRequirement) that determines the low flow releases from each reservoir. Low flow releases will be limited to this value.
I/O: Required input
Links: Not linkable

21.1.23 Sediment

The Sediment category is used to enable algorithms which adjust reservoir Elevation Volume and possibly Elevation Area relationships in response to sediment inflow.

21.1.23.1 None

The None method is the default for the Sediment category. No calculations are performed in this method. There are no slots specifically associated with this method.

21.1.23.2 CRSS Sediment

The **CRSS Sediment** method is designed based on sedimentation calculations performed by the US Bureau of Reclamation’s Colorado River Simulation System (CRSS) model. This function distributes reservoir sediment based on the “Empirical Area Reduction Method”. Simply put, sediment is distributed through an iterative process in which a total volume loss due to sedimentation is calculated based on an assumed top of sediment elevation.

Slots Specific to this Method

☛ **Elevation Area Table**

Type: Table
Units: LENGTH vs. AREA
Description: generated elevation area table for calculating sediment distribution
Information:
I/O: Output only
Links: Not linkable

☛ **Elevation Vol_Area Table Increment**

Type: Table
Units: LENGTH
Description: elevation increments for the generated Elevation Volume and Elevation Area Tables
Information: This table often needs more precise elevation increments than the sediment calculation tables.
I/O: Required input
Links: Not linkable

☛ **Initial Elevation Area Table**

Type: Table
Units: LENGTH vs. AREA
Description: initial elevation area table

Information: Provided for comparison with initial data
I/O: Output only
Links: Not linkable

Initial Elevation Volume Table

Type: Table
Units: LENGTH vs. VOLUME
Description: initial elevation volume table
Information: provided for comparison with initial data
I/O: Output only
Links: Not linkable

Sediment Distribution Coefficients

Type: Table
Units: NOUNITS
Description: parameters for empirical equation governing sediment distribution
Information:
I/O: Required input
Links: Not linkable

Sediment Inflow

Type: Series
Units: VOLUME
Description: volume of sediment flowing into the reservoir at each timestep
Information:
I/O: Required input
Links: Not linkable

User Input Elev Area Data

Type: Table
Units: LENGTH vs. AREA
Description: initial Elevation Area relationship
Information: These values are initial conditions for the first timestep of the simulation. The elevation increments will be used for all sedimentation calculations.
I/O: Required input
Links: Not linkable

This volume loss is recalculated (with a new top of sediment elevation) at each iteration, until the calculated volume loss is equal to the actual volume of sediment inflow (within a specified convergence). The total volume loss calculation consists of a somewhat complicated algorithm utilizing elevation/area and elevation/volume data for the reservoir and an empirical equation. The empirical equation uses user specified parameters which relate the portion of total area that is taken up by sediment to the Pool Elevation. The empirical equation basically gives the shape of the accumulated sediment. The empirical equation has a close relationship to the elevation volume and elevation area characteristics of a given reservoir. The elevation/area and elevation/volume data is stored in a polynomial coefficient table, which gets recalculated after each timestep. The actual Elevation Area, Elevation Volume tables used by **RiverWare**™ are adjusted at the end of the sedimentation code (but prior to the hydrologic simulation).

Caution should be exercised in creating input data for this method. The close relationship between the empirical area reduction equation and the shape of the reservoir (reflected in the User Input Elev Area Data) makes the method fairly sensitive to input data. When choosing empirical parameters for this method, physical characteristics of the given reservoir need to be considered. The Bureau of Reclamation currently considers 4 possible types of reservoirs, with each type having a corresponding set of empirical area reduction parameters. The reservoir type classification is based on the shape of the Reservoir, the manner in which the reservoir is to be operated, and the size of the sediment particles to be deposited in the reservoir. The main emphasis is on the shape. Tables are used to classify the reservoirs based on these characteristics. Once the type has been established, the parameter values for that type can also be taken from tables in the literature. An incorrect set of parameters for a given reservoir will lead to an inability to achieve convergence on the sediment distribution within this method.

21.1.24 Bank Storage

The Bank Storage methods are used to calculate the volume of water stored in the Reservoir banks. These methods also calculate the change in the volume of water stored in the Reservoir banks from one timestep to the next.

21.1.24.1 None

None should be chosen if the user does not want to calculate the amount of Bank Storage in the Reservoir. This is the default method for the Bank Storage category. Bank Storage and the Change in Bank Storage are set to zero but are not displayed. There are no slots specifically associated with this method. No calculators are performed in this method.

21.1.24.2 Input Bank Storage

The Input Bank Storage method allows users to directly input values into the Bank Storage slot or to set these values using a rule. Change in Bank Storage is calculated internally in RiverWare for use in the mass balance equations.

Slots Specific to this Method

☛ **Bank Storage**

Type: Series Slot
Units: Volume
Description: volume of water stored in the reservoir banks
Information:
I/O: Input Only
Links: Usually not linked, but could be linked to Data Object.

☛ **Change in Bank Storage**

Type: Series Slot
Units: Volume
Description: change in volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

21.1.24.3 CRSS Bank Storage

The CRSS Bank Storage method replicates the U.S. Bureau of Reclamation’s CRSS bank storage calculation. The Bank Storage and the Change in Bank Storage are calculated using the Reservoir Storage and the Bank Storage Coefficients.

Slots Specific to this Method

☛ Bank Storage

Type: Series Slot
Units: Volume
Description: volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

☛ Bank Storage Coefficient

Type: Table Slot
Units: No Units vs. No Units
Description: gain or loss of storage vs. change in bank storage
Information: The first coefficient (column zero) is for increasing storage and the second coefficient is for decreasing storage.
I/O: Required input
Links: Not linkable

☛ Change in Bank Storage

Type: Series Slot
Units: Volume
Description: change in volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

There are two ways Bank Storage can be calculated depending on the current Storage of the Reservoir. If the Reservoir's current Storage is greater than the Reservoir's Storage at the previous timestep, the Storage is increasing. Bank Storage is calculated using the following equation:

$$\text{Bank Storage} = \text{Bank Storage(-1)} + (\text{first Bank Storage Coefficient} \times (\text{Storage} - \text{Storage(-1)}))$$

If the Reservoir's current Storage is less than the Reservoir's Storage at the previous timestep, the Storage is decreasing. Bank Storage is calculated using the following equation:

$$\text{Bank Storage} = \text{Bank Storage(-1)} + (\text{second Bank Storage Coefficient} \times (\text{Storage} - \text{Storage(-1)}))$$

The Change in Bank Storage is calculated using the following equation regardless of which method was used to compute Bank Storage.

$$\text{Change in Bank Storage} = \text{Bank Storage} - \text{Bank Storage}(-1)$$

where in the above equations:

Bank Storage = the volume of water stored in the banks of the Reservoir at the current timestep

Bank Storage(-1) = the volume of the water stored in the banks of the Reservoir at the previous timestep.

Storage = the volume of water in the Reservoir at the current timestep

Storage(-1) = the volume of water in the Reservoir at the previous timestep

21.1.24.4 Average Stage Change

The Average Stage Change method calculates the Bank Storage and Change in Bank Storage based on the flow from storage. The flow from storage is a function of the average stage change over a user defined number of timesteps.

Slots Specific to this Method

☛ Ave Stage Change Coeffs

Type: Table Slot
Units: Area per Time and Flow
Description: coefficient describing flow for a given change in pool elevation and a constant representing flow from bank storage
Information:
I/O: Required input
Links: Not linkable

☛ Bank Storage

Type: Series Slot
Units: Volume
Description: volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

☛ Change in Bank Storage

Type: Series Slot
Units: Volume
Description: change in volume of water stored in the reservoir banks
Information:

I/O: Output only

Links: Not linkable

☛ Timesteps to Average

Type: Table Slot

Units: No Units

Description: number of timesteps used to calculate average pool elevation.

Information:

I/O: Required input

Links: Not linkable

The average stage change is calculated using the following equation:

$$\text{average Pool Elevation} = \frac{\text{Pool Elevation} - \text{Pool Elevation (-Timesteps to Average)}}{\text{Timesteps to Average}}$$

The change in flow to bank storage is calculated using the following equation:

$$\text{Flow to banks} = \text{Average Stage Change Bank Storage Coefficient} \times \text{Average Pool Elevation} + \text{Average Stage Change Bank Storage Constant}$$

The flow is converted to a volume by multiplying the value by the current timestep. The Change in Bank Storage is calculated using the following equation:

$$\text{Change in Bank Storage} = \text{Bank Storage} - \text{Bank Storage(-1)}$$

21.1.25 Diversion from Reservoir

The Diversion from Reservoir user methods are applicable when a reservoir is linked to a diverting object (e.g. AggDiversionSite, AggDistributionCanal, or Diversion Object). These methods simply create the slots which must be linked (by the user) to slots on the diverting object.

21.1.25.1 None

This is the default for the Diversion from Reservoir category. It is used when the reservoir is not linked to a diverting object. If the reservoir is linked to a diverting object and this method is selected, the object will not solve correctly. There are no slots specifically associated with this method.

21.1.25.2 Available Flow Based Diversion

This method must be selected when a reservoir is linked to either an AggDiversionSite, AggDistributionCanal, or a Diversion Object that is using the Available For Diversion Linked method. Selecting this method allows the Available for Diversion slot to be available for linking. The AggDiversionSite, AggDistributionCanal, and Diversion objects contain more information about diverting water from a reservoir.

SLOTS SPECIFIC TO THIS METHOD

☛ **AVAILABLE FOR DIVERSION**

Type: Series
Units: FLOW
Description: represents the amount of water that may be diverted from the reservoir
Information:
I/O: Optional; can be input by the user or determined by **RiverWare**™.
Links: Should be linked to the Available for Diversion slot on AggDiversionSite or Diversion object, or the Incoming Available Water slot on a Water User.

Available for Diversion can either be input by the user or calculated by the reservoir. If it is not input it is set as the previous Storage divided by the timestep length. The value is limited to not be negative.

No other calculations are performed if this method is selected.

21.1.25.3 Head Based Diversion

This method may be selected when a reservoir is linked to a Diversion Object. Selecting this method allows the Previous Pool Elevation slot to be available for linking. The Diversion Object contains more information about diverting water from a Reservoir.

Slots Specific to this Method

☛ Previous Pool Elevation

Type: Series
Units: Length
Description: Pool Elevation value for the previous timestep
Information:
I/O: Output only
Links: Should be linked to the Diversion Intake Elevation slot on the Diversion Object.

21.1.26 Diversion Power

The methods in this category calculate power generated on the diversion from the reservoir. Selecting a method other than None in this category will make the Diversion Tailwater and Diversion Power Bypass categories available. If a method other than None is selected for Diversion Power, then a method other than None must be selected for Diversion Tailwater.

21.1.26.1 None

This is the default method for the Diversion Power category. No calculations are performed in this method, and there are no slots specifically associated with this method.

21.1.26.2 Diversion Power Efficiency Curve

The Diversion Power Efficiency Curve method is similar to the Plant Efficiency Curve method in the Power category with the exception that the method does not allow **Diversion Energy** to input or set by rules (nor can **Diversion Energy** be set with the Best Efficiency or Max Capacity flags). **Diversion Energy** and **Diversion Power** are only calculated as outputs. The method calculates **Diversion Power** by a 3-D interpolation of the **Diversion Power Table** using the current, average **Diversion Operating Head** and **Diversion Turbine Flow**. The **Diversion Power Coefficient** is calculated as **Diversion Power** divided by **Diversion Turbine Flow**. Alternatively, the user can input **Diversion Power Coefficient**, and then **Diversion Power** is calculated directly as the **Diversion Power Coefficient** multiplied by the **Diversion Turbine Flow**.

Slots Specific to this Method

☛ **Diversion Power Table**

Type: Table Slot
Units: Length vs Flow vs Power
Description: 3-D table representing the power characteristics of the diversion power plant, used to calculate power using interpolation
Information: Data must be entered into the table in increasing, blocks of the same Diversion Operating Head. For every block of the same Diversion Operating Head in column 1, Diversion Turbine Flow should be listed in increasing order in column 2 and the corresponding Diversion Power in column 3. The first row for each Diversion Operating Head must be for zero Diversion Turbine Flow and zero Diversion Power.
I/O: Required input
Links: Not linkable

Div Head	Turbine Flow	Div Power
30	0	0
30	100	100
30	200	175
40	0	0
40	100	125
40	220	195
50	0	0
50	110	147
50	250	205

☛ Diversion Max Turbine Table

Type: Table Slot
Units: Length vs Flow
Description: The maximum Diversion Turbine Flow as a function of Diversion Operating Head
Information: RiverWare automatically populates this table at the start of the run using the Diversion Power Table. The first column contains the Diversion Operating Head values from the Diversion Power Table, one row for each unique Diversion Operating Head in increasing order. The second column contains the maximum Diversion Turbine Flow value for each Diversion Operating Head.
I/O: Output only
Links: Not Linkable

☛ Diversion Power Cap Fraction

Type: Series Slot
Units: Fraction
Description: This is the percentage of full capacity of the turbine units in the diversion power plant. For example, if only half of the turbine are operational (and they are all the same), this value would be 0.5.
Information: This must be a number between 0 and 1 (inclusive). If not input or set by rules, this slot is automatically set to 1.
I/O: Optional input, if not, value is set to 1
Links: Not linkable

☛ Diversion Operating Head

Type: Series Slot

Units: LENGTH
Description: The difference between the average Pool Elevation and the Diversion Tailwater Elevation
Information:
I/O: Output only
Links: Not usually linked

DIVERSION TURBINE FLOW

Type: Series Slot
Units: FLOW
Description: The diversion flow that passes through the turbines to generate power
Information: If the slot is not input or set by rules, then it is calculated as the difference between Diversion and Diversion Power Bypass if Diversion Power Bypass is input or set by rules. If neither Diversion Turbine Flow nor Diversion Power Bypass is input or set by rules, then Diversion Turbine Flow is calculated as the lesser of Diversion and the calculated maximum diversion turbine flow based on the Diversion Max Turbine Table and the current Diversion Operating Head. It is not permissible to have both Diversion Turbine Flow and Diversion Power Bypass as input or set by rules.
I/O: Optional input or output
Links: Not linkable

DIVERSION POWER

Type: Series Slot
Units: POWER
Description: The power generated from flow through the reservoir diversion
Information: If Diversion Power Coefficient is not input or set by rules, Diversion Power is calculated using a 3-D interpolation on the Diversion Power Table given the current, average Diversion Operating Head and the current Diversion Turbine Flow, scaled by the Diversion Power Cap Fraction. If Diversion Power Coefficient is input or set by rules, Diversion Power is calculated as the Diversion Power Coefficient multiplied by the Diversion Turbine Flow.
I/O: Output only
Links: Linkable

DIVERSION ENERGY

Type: Series Slot
Units: ENERGY
Description: The energy generated from flow through the reservoir diversion
Information: Calculated as the Diversion Power multiplied by the timestep length
I/O: Output only

Links: Linkable

☛ Diversion Power Coefficient

Type: Series Slot
Units: Power per Flow
Description: The power generation per unit of flow through the turbines on the reservoir diversion
Information: If this slot is input or set by rules, it is used directly to calculate Diversion Power. If it is not input or set by rules, then it is calculated as Diversion Power divided by Diversion Turbine Flow. If either Diversion Power, Diversion Turbine Flow or Diversion Power Cap Fraction is zero, then this slot will be zero.
I/O: Optional input or output
Links: Not usually linked

At the start of the run, the **Diversion Max Turbine Table** slot is populated using the **Diversion Power Table**. The first column is populated with each unique **Diversion Operating Head** value from the **Diversion Power Table**, in ascending order. The second column is populated with the corresponding maximum diversion turbine flow value.

When the method executes, **Diversion** will already be known. The method calls the selected Diversion Tailwater method to calculate the **Diversion Tailwater Elevation**. If the default method, **None**, is selected for the **Diversion Tailwater** category, the run will abort and an error message will be issued.

The method then calculates the **Diversion Operating Head**:

$$\text{Diversion Operating Head}[t] = \frac{\text{Pool Elevation}[t-1] + \text{Pool Elevation}[t]}{2} - \text{Diversion Tailwater Elevation}[t]$$

The method calculates the *maxDiversionTurbineFlow* by interpolating the **Diversion Max Turbine Table** slot using the **Diversion Operating Head**. The value is scaled by the **Diversion Power Cap Fraction**.

If **Diversion Turbine Flow** is specified (input or set by rules) it is checked against the *maxDiversionTurbineFlow*, and if the specified value exceeds the max, the run will abort with an error message. Otherwise a temporary turbine flow is calculated.

$$\text{tempDiversionTurbine} = Min(\text{Diversion} - \text{Diversion Power Bypass}, maxDiversionTurbineFlow)$$

If **Diversion Power Bypass** is not input or set by rules, or if the **Diversion Power Bypass** method is **None**, the **Diversion Power Bypass** will have defaulted to zero at this point.

If the combined temporary turbine flow plus the current **Diversion Power Bypass** is less than the (total) **Diversion**, then the method then calls the selected Diversion Power Bypass method to increase the **Diversion Power Bypass** to make up the difference. If it is not possible for the turbine flow plus the bypass to equal the total **Diversion**, either due to the values being specified (input or rules) or due to max capacity limits, then the run will abort with an error message.

The method then sets the Diversion Turbine Flow slot:

$$\text{Diversion Turbine Flow} = \text{tempDiversionTurbine}$$

If **Diversion Power Coefficient** is specified (input or rules), it is used to calculate **Diversion Power** directly:

$$\text{Diversion Power} = \text{Diversion Turbine Flow} \times \text{Diversion Power Coefficient}$$

Otherwise, **Diversion Power** is calculated by a 3-D interpolation on the **Diversion Power Table** using **Diversion Operating Head** and **Diversion Turbine Flow**, and **Diversion Power Coefficient** is calculated as:

$$\text{Diversion Power Coefficient} = \frac{\text{Diversion Power}}{\text{Diversion Turbine Flow}}$$

Diversion Energy is then calculated as **Diversion Power** multiplied by the timestep length.

Notes on Diversion Power Cap Fraction

If the **Diversion Power Cap Fraction** is input by the user, it is necessary for the **Diversion Power Table** to be scaled back to account for the operating points when the turbines are operating at less than 100%. To do this, **Diversion Turbine Flow** is divided by the **Diversion Power Cap Fraction**. This point is then found in the **Diversion Power Curve** for the current **Diversion Operating Head**, and the power is found using 3-D interpolation. Finally the power is multiplied by the **Diversion Power Cap Fraction** to get the actual **Diversion Power** produced for the current timestep.

21.1.27 Diversion Tailwater

The methods in this category calculate the elevation of the tailwater on the diversion from a reservoir. This category is dependent on the selection of a method other than the default method, None, in the Diversion Power category. If a method other than the default is selected for Diversion Power, then a method other than the default, None, must be selected for Diversion Tailwater.

21.1.27.1 None

This is the default method for the Diversion Tailwater category. No calculations are performed in this method, and there are no slots specifically associated with this method.

21.1.27.2 Diversion Base Value Plus Lookup

The Diversion Base Value Plus Lookup method computes the **Diversion Tailwater Elevation** by added the average **Diversion Tailwater Base Value** (over the timestep) to a function of **Diversion** defined in the **Diversion Tailwater Table** slot. This method is similar to the Base Value Plus Lookup Table method in the Tailwater category but uses the **Diversion** and **Diversion Tailwater Base Value** slots instead of **Outflow** and **Tailwater Base Value**. The **Diversion Tailwater Base Value** may be input by the user or linked to another slot, such as the **Pool Elevation** of another Reservoir. If the **Tailwater Base Value** is neither input nor linked, it is automatically set to zero.

Slots Specific to this Method

☛ Diversion Tailwater Table

Type: Table Slot
Units: Flow vs Length
Description: This slot defines the relationship between Diversion and the Diversion Tailwater Elevation; Diversion vs either the diversion tailwater elevation or the tailwater elevation increment
Information: If the Diversion Tailwater Base Value is non-zero, the Diversion Tailwater Table gives values of incremental increase in Tailwater Elevation over th base value. Otherwise, the table gives the Diversion Tailwater Elevation values. The first row of the table should be for a Diversion flow of zero.
I/O: Required input
Links: Not linkable

☛ Diversion Tailwater Base Value

Type: Series Slot
Units: Length
Description: the base elevation of the diversion tailwater, such as a downstream stage

Information: If the slot is not input or linked, it defaults to 0.
I/O: Optional, can be input or linked
Links: Linkable

DIVERSION TAILWATER ELEVATION

Type: Series Slot
Units: LENGTH
Description: the water surface elevation of the tailwater from the reservoir diversion
Information: This slot is used to compute Diversion Operating Head in Diversion Power calculations
I/O: Output only
Links: Not linkable

When this method is executed, the **Diversion** value will already be known. If the **Diversion Tailwater Base Value** is neither linked, input nor set by rules, then it will default to zero.

The following steps are performed to calculate **Diversion Tailwater Elevation**.

1. *TWTemp* is obtained from a table interpolation on the **Diversion Tailwater Table** using **Diversion**.

2. If both **Diversion Tailwater Base Value[t]** and **Diversion Tailwater Base Value[t-1]** are known, then the **Diversion Tailwater Elevation** is calculated as:

$$\text{Diversion Tailwater Elevation}[t] = \frac{\text{Diversion Tailwater Base Value}[t-1] + \text{Diversion Tailwater Base Value}[t]}{2} + TWTemp$$

3. If **Diversion Tailwater Base Value[t]** is known, but **Diversion Tailwater Base Value[t-1]** is not known, then the **Diversion Tailwater Elevation** is calculated as:

$$\text{Diversion Tailwater Elevation}[t] = \text{Diversion Tailwater Base Value}[t] + TWTemp$$

4. If **Diversion Tailwater Base Value[t-1]** is known, but **Diversion Tailwater Base Value[t]** is not known, then the **Diversion Tailwater Elevation** is calculated as:

$$\text{Diversion Tailwater Elevation}[t] = \text{Diversion Tailwater Base Value}[t-1] + TWTemp$$

5. If neither **Diversion Tailwater Base Value[t]** nor **Diversion Tailwater Base Value[t-1]** are known but **Diversion Tailwater Elevation[t-1]** is known, the current timestep's **Diversion Tailwater Elevation** is set equal to **Diversion Tailwater Elevation[t-1]**.

6. If neither **Diversion Tailwater Base Value[t]**, **Diversion Tailwater Base Value[t-1]**, nor **Diversion Tailwater Elevation[t-1]** are known, or if **Diversion** is not known, the method will exit and wait for more information.

21.1.28 Diversion Power Bypass

The methods in this category calculate the portion of the diversion from a reservoir that does not pass through the turbines but rather through a bypass structure. This category is dependent on the selection of a method other than the default method, None, in the Diversion Power category.

21.1.28.1 None

This is the default method for the Diversion Power Bypass category. No calculations are performed in this method, and there are no slots specifically associated with this method. If this method is selected, it is assumed that all **Diversion** flow passes through the turbines.

21.1.28.2 Bypass Capacity Table

This method sets **Diversion Power Bypass** to the difference between **Diversion** and **Diversion Turbine Flow** if it is not input or set by rules, and it checks that the **Diversion Power Bypass** does not exceed the maximum based on the **Diversion Power Bypass Table**. This functions similarly to the Regulated method in the Spill category.

Slots Specific to this Method

☛ **Diversion Power Bypass Table**

Type: Table Slot
Units: Length vs Flow
Description: Pool Elevation vs. the corresponding maximum diversion power bypass values
Information:
I/O: Required input
Links: Not linkable

☛ **Diversion Power Bypass**

Type: Series
Units: Flow
Description: Diversion flow that does not pass through power turbines
Information: If not input or set by rules, Diversion Power Bypass will be set equal to the difference between Diversion and Diversion Turbine Flow.
I/O: Optional input or output
Links: Linkable

At the beginning of the run, if **Diversion Power Bypass** is not specified (input or rules), it is initially set to a default value of zero.

On each timestep, the method first checks if both **Diversion Power Bypass** and **Diversion Turbine Flow** are input or set by rules. If so, the run will abort with an error message. It necessary to leave at least one of these slots as a free variable.

The method then calculates max diversion power bypass by performing a table interpolation on the **Diversion Power Bypass Table** using the average **Pool Elevation** from the end of the current timestep and end of the previous timestep.

If **Diversion Power Bypass** is input or set by a rule, then the value is checked against the max diversion power bypass, and if it exceeds the max, the run will abort with an error message.

If **Diversion Power Bypass** is not input or set by a rule, then it is calculated as:

$$\text{Diversion Power Bypass} = \text{Diversion} - \text{Diversion Turbine Flow}$$

The calculated **Diversion Power Bypass** value is checked against the max diversion power bypass, and if it exceeds the max, the run will abort with an error message.

21.1.29 Seepage

The Seepage methods are used to calculate the amount of water lost through the face of the dam. The volume of seepage computed during the execution of these methods affects the mass balance of the Reservoir.

21.1.29.1 None

None is the default for the Seepage category. It is used when the user does not want to calculate the flow of water through the face of the dam.

21.1.29.2 Input Seepage

The Input Seepage method is used when it is desired to have the seepage slot as input or set by a rule.

SLOTS ADDED BY THIS METHOD:

☛ **SEEPAGE**

Type: SeriesSlot
Units: FLOW
Description: flow of water through the dam face
Information: Seepage is not included in the Outflow of the reservoir and will need to be linked separately if the water does in fact go downstream.
I/O: Input only
Links: Linkable

21.1.29.3 Linear Seepage

The Linear Seepage method calculates the seepage from the face of the dam. This calculation is based on the Pool Elevation of the Reservoir and specified coefficients.

SLOTS ADDED BY THIS METHOD:

☛ **SEEPAGE**

Type: SeriesSlot
Units: FLOW
Description: flow of water through the dam face
Information: Seepage is not included in the Outflow of the reservoir and will need to be linked separately if the water does in fact go downstream.
I/O: Output only
Links: Linkable

☛ **Seepage Coefficients**

Type: TableSlot
Units: LENGTH, AREA PER TIME, FLOW
Description: coefficients in the linear equation for seepage
Information: The first coefficient (column zero) is the base elevation of the dam. The second coefficient is the slope of the linear equation for seepage. The third coefficient is the intercept of the linear equation for seepage.
I/O: Required Input
Links: NA

The calculation for Seepage in this method is fairly straightforward. A linear model is used. The coefficient are user inputs. The following equation is used to compute Seepage:

$$\text{Seepage} = (\text{Pool Elevation} - \text{first Seepage Coefficient}) \times \text{second Seepage Coefficient} + \text{third Seepage Coefficient}$$

21.1.29.4 Single Seepage Value

The Single Seepage Value method sets the seepage from the face of the dam equal to a scalar value.

Slots added by this method:

☛ **Seepage**

Type: Series
Units: Flow
Description: Flow of water through the dam face
Information: Seepage is not included in the Outflow of the reservoir and will need to be linked separately if the water does in fact go downstream.
I/O: Optional Input
Links: Linkable

☛ **Single Seepage Value**

Type: Scalar
Units: Flow
Description: seepage value to be applied to each timestep
Information:
I/O: Required Input
Links: Not Linkable

This method is executed at the beginning of the run. For each timestep from the initial timestep through the end of the run (plus post run dispatching timesteps too), if the **Seepage** is not input, the **Seepage** is set equal to the **Single Seepage Value**.

The method will issue an error if there is not a valid value in the **Single Seepage Value** slot. Note, this structure allows some flexibility. Seepage can be input/rules when necessary but will use the scalar value when not input.

21.1.29.5 Linked Seepage

This method is intended to be used when linking a Reservoir object with a Groundwater Storage object that uses the **Head Based Boundary Condition** method in the **Solution Type** category **HERE (Objects.pdf, Section 14.1.1.3)**.

SLOTS ADDED BY THIS METHOD:

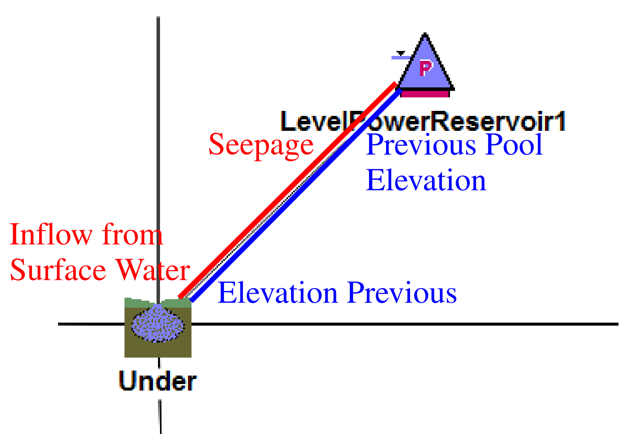
☛ **SEEPAGE**

Type: Series
Units: FLOW
Description: Flow of water out of the reservoir, often into groundwater
Information: A positive value is flow out of the reservoir.
I/O: Output only if linked to a Groundwater object (typical); otherwise required input
Links: Must be linked, typically to **Inflow from Surface Water** on Groundwater object

☛ **PREVIOUS POOL ELEVATION**

Type: Series
Units: LENGTH
Description: Pool Elevation at the end of the previous timestep
Information:
I/O: Output only
Links: Typically linked to the **Elevation Previous** slot on a Groundwater object

This method does not do any calculations; it just adds the appropriate slots. The Reservoir does provide the **Previous Pool Elevation** which is then linked to the Groundwater object **Elevation Previous**. The Groundwater object computes the **Inflow from Surface Water**, which is linked back to the **Seepage** slot on the reservoir. The **Seepage** is used in the Reservoir mass balance.



Click **HERE (Objects.pdf, Section 14.1.1.3)** for more information about how **Seepage** is calculated as **Inflow from Surface Water** on the linked Groundwater object.

21.1.30 Operating Levels

This category enables the user to specify operating levels for the reservoir. Operating levels serve as a normalizing metric for reservoir contents. This metric is used by reservoir-balancing algorithms to determine the relative "fullness" of reservoirs. On individual reservoirs, it also serves to identify the elevations that correspond to pool boundaries, such as the top of the conservation pool or the top of the flood pool.

21.1.30.1 None

This is the default method; no slots are instantiated and no calculations are performed.

21.1.30.2 Conservation Pools

This method allows the user to specify that there is a conservation pool for this reservoir.

SLOTS SPECIFIC TO THIS METHOD

☛ **OPERATING LEVEL TABLE**

Type: PeriodicSlot
Units: TIME VS LENGTH AT OPERATING LEVELS
Description: table describing the seasonal variation of elevation in a reservoir at each of the user-designated operating levels.
Information: number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep.
I/O: Required Input
Links: Not Linkable

☛ **OPERATING LEVEL**

Type: SeriesSlot
Units: NONE
Description: The computed operating level
Information: This slot is computed using the pool elevation and the Operating Level Table
I/O: Output only
Links: Not Linkable

☛ Operating Level Storage Table

Type: PeriodicSlot
Units: Time vs Volume at Operating Levels
Description: table describing the seasonal variation of storage in a reservoir at each of the user-designated operating levels.
Information: Each column represents the time-varying storage for a particular Operating Level. The integer value of the Operating Level is in the header of each column. This table is generated from the Operating Level Table and has the same number of rows and columns. The values in the table are storage values (looked up from the elevation volume table) whereas the values in the Operating Level Table are elevations. This slot is computed at run-time so it is read-only to the user. All changes should be made in the Operating Level Table.
I/O: Output Only
Links: Not Linkable

☛ Top of Conservation Pool

Type: ScalarSlot
Units: NONE
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the conservation pool.
Information:
I/O: Input
Links: Not Linkable

☛ Bottom of Conservation Pool

Type: ScalarSlot
Units: NONE
Description: Operating level (as defined in Operating Level Table) corresponding to the bottom of the conservation pool.
Information: Used by some conservation pool operations algorithms, along with the Top of Conservation Pool slot, to identify the volume in the conservation pool.
I/O: Input Only
Links: Not Linkable

☛ Conservation Pool Initial Empty Space

Type: SeriesSlot
Units: None
Description: The inflow required to fill the conservation pool at the beginning of timestep, based on the ending storage at the prior timestep, taking into account evaporation, precipitation, etc.

Information: This slot is computed at the beginning of the timestep; evaporation rates and other such factors that are not already defined at the beginning of timestep will not be taken into account in this computation. This slot is used by the water rights allocation solution algorithm to compute physical constraints and by storage accounts to compute allocation requests.
I/O: Output only
Links: Not Linkable

☛ Conservation Pool Storage

Type: Series
Units: Volume
Description: This is the computed volume of water in the conservation pool.
Information: This value is always non-negative.
I/O: Output only
Links: Not Linkable

☛ Conservation Pool Full Storage

Type: Series
Units: Volume
Description: This is the possible volume of water that could be stored in the Conservation pool. It is computed as the Storage at the top of the conservation pool minus the storage at the bottom of the conservation pool
Information:
I/O: Output Only
Links: Not Linkable

☛ Conservation Pool Storage Fraction

Type: Periodic Slot
Units: Fraction
Description: The values in the periodic slot represent the percentage of the conservation pool storage at each level (column) in the Operating Level Table.
Information: It has identical dimension including dates and levels as the Operating Level Table. This table will be populated at beginning of run. The Operating Level Table will be its “source” slot.
I/O: Output only
Links: Not Linkable

At the beginning of run, the Conservation Pool Storage Fraction is populated as follows: For each date (row) and each level, n, (column), the equation to compute the fraction:

$$\text{Conservation Pool Storage Fraction}[date,n]=$$

$$\frac{\text{Op Level Storage Table}[date, n] - \text{Op Level Storage Table}[date, \text{Bottom of Cons Pool}]}{\text{Op Level Storage Table}[date, \text{Top of Cons Pool}] - \text{Op Level Storage Table}[date, \text{Bottom of Cons Pool}]}$$

Note, the Conservation Pool Storage Fraction is limited to be always between 0 and 1 (0% to 100%).

At the end of each dispatch method, the Operating Level series slot is computed by looking up the pool elevation and date on the Operating Level Table.

Next, **Conservation Pool Full Storage** is calculated as follows.

$$\text{Conservation Pool Full Storage}[t] = \text{Operating Level Storage Table}[t, \text{Top of Conservation Pool}] - \text{Operating Level Storage Table}[t, \text{Bottom of Conservation Pool}]$$

The **Conservation Pool Storage** is computed as:

If the Operating Level is greater than the Top of the Conservation Pool,

$$\text{Conservation Pool Storage}[t] = \text{Full Conservation Pool Storage}[t]$$

else if the Operating Level is less than the Bottom of the Conservation Pool,

$$\text{Conservation Pool Storage}[t] = 0$$

else

$$\text{Conservation Pool Storage}[t] = \text{Storage}[t] - \text{Operating Level Storage Table}[t, \text{Bottom of Conservation Pool}]$$

21.1.30.3 Conservation and Flood Pools

This method allows the user to specify that there is a conservation and a flood pool for this reservoir.

Slots Specific to this Method

This method is an extension of the Conservation Pool method, and selecting this method causes all the slots for Conservation Pool to become available, along with the following:

☛ Top of Flood Pool

Type: ScalarSlot
Units: No Units
Description: Operating level (as defined in Operating Level Table) corresponding to the top of flood pool.

Information:
I/O: Required Input
Links: Not Linkable

☛ Flood Pool Storage

Type: Series
Units: Volume
Description: This is the computed volume of water in the flood pool.
Information: This value is always non-negative.
I/O: Output Only
Links: Not Linkable

☛ Flood Pool Full Storage

Type: Series
Units: Volume
Description: This is the possible volume of water that could be stored in the Flood pool. It is computed as the Storage at the top of the flood pool minus the storage at the top of the conservation pool.
Information:
Links: Not Linkable

☛ Flood Pool Storage Fraction

Type: Periodic Slot
Units: Fraction
Description: The values in the periodic slot represent the percentage of the flood pool storage at each level (column) in the Operating Level Table.
Information: It has identical dimension including dates and levels as the Operating Level Table. This table will be populated at beginning of run. The Operating Level Table will be its “source” slot.
I/O: Output only
Links: Not Linkable

At the beginning of run, the Conservation Pool Storage Fraction is populated as described above. Then, the Flood Pool Storage Fraction is populated as follows: For each date (row) and each level, n, (column), the equation to compute the fraction is:

$$\text{Flood Pool Storage Fraction}[date,n] = \frac{\text{Op Level Storage Table}[t, n] - \text{Op Level Storage Table}[t, \text{Top of Cons Pool}]}{\text{Op Level Storage Table}[t, \text{Top of Flood Pool}] - \text{Op Level Storage Table}[t, \text{Top of Cons Pool}]}$$

Note, the Flood Pool Storage Fraction is limited to be always greater than 0. But, it can be larger than 1 (100%). For levels above the flood pool, the percentage will be greater than 100%.

At the end of each dispatch method, the Operating Level series slot is computed by looking up the pool elevation and date on the Operating Level Table. Next, all slots associated with the Conservation Pool are computed and set as described above. Then, **Flood Pool Full Storage** is calculated as follows:

$$\text{Flood Pool Full Storage}[t] = \text{Operating Level Storage Table}[t, \text{Top of Flood Pool}] - \text{Operating Level Storage Table}[t, \text{Top of Conservation Pool}]$$

The **Flood Pool Storage** is computed as:

If the Operating Level is less than the Top of the Conservation Pool,

$$\text{Flood Pool Storage}[t] = 0$$

else

$$\text{Flood Pool Storage}[t] = \text{Storage}[t] - \text{Operating Level Storage Table}[t, \text{Top of Conservation Pool}]$$

Note, the **Flood Pool Storage** may be larger than the **Flood Pool Full Storage**. This indicates the reservoir is above the flood pool and is surcharging.

21.1.31 Conditional Operating Levels

This category provides methods that allows the user to use alternative operating level tables based on conditions in the run.

21.1.31.1 None

This is the default method; no slots are instantiated and no calculations are performed. The original Operating Level Table is used for all computations.

21.1.31.2 Sum Inflows over Interval

This method allows an alternative operating level table (i.e. a guide curve) to be used starting on a certain date if a certain combination of flows are high enough for a specified time range.

For example, if there has been a total of 200,000 acre-feet of total inflows into a specific reservoir during the months of March, April, and May, then on June 15th, the method would switch the reservoir operations to follow an alternative table. On October 15th, the reset date, the reservoir will once again use the original Operating Level Table.

Slots Specific to this Method

☛ **Operating Level 2 Table**

Type: Periodic
Units: Time vs Length at Operating Levels
Description: This is the alternative operating level table that is used when indicated by hydrologic conditions. This table describes the seasonal variation of elevation in a reservoir at each of the user-designated operating levels.
Information: Number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep. This table should have the same dimensions (rows and columns) as the Operating Level Table.
I/O: Required Input
Links: Not Linkable

☛ **Operating Level 2 Trigger Volume**

Type: Scalar
Units: Volume

Description: The volume of inflows (**Inflow Sum** slot) between the criteria start and finish (specified on the **Operating Level 2 Dates** slot) that causes the reservoir to use the **Operating Level 2 Table** rather than the original **Operating Level Table.**

I/O: Required Input or an error will be issued at the start of run.

Links: NA

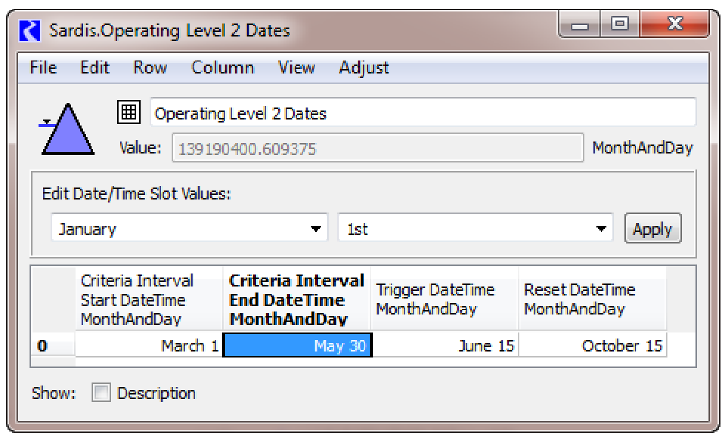
OPERATING LEVEL 2 DATES

Type: Table

Units: DATETIME

Description: This table has 4 columns. The first and second columns are the “Criteria Interval Start DateTime” and “Criteria Interval End DateTime” between which the flow volume is summed and compared to the **Operating Level 2 Trigger Volume**. The third column is the “Trigger DateTime” at which the reservoir will check the conditions and possibly use the **Operating Level 2 Table** rather than the original **Operating Level Table.** The fourth column is the “Reset DateTime” at which the reservoir will use the **Operating Level Table** again.

Information: The units for this slot are DateTime which can be an absolute or partially specified datetime. The default user units are “MonthAndDay”. Partially specified datetimes are converted to fully specified datetimes using information from the current timestep to fill in the missing pieces of the partially specified timestep. Thus, if the datetime is partially specified, it must be able to evaluate to a timestep in the model or an error will be issued.



I/O: Required input or an error will be issued at the start of run.

Links: NA

OPERATING LEVEL STORAGE 2 TABLE

Type: Periodic

Units: STORAGE

Description: This represents the Storage associated with the elevations in the Operating Level 2 Table.

Information: This slot is created at the beginning of run. The Operating Level 2 Table is its “source” slot.
I/O: Output Only
Links: Not Linkable.

☛ Conservation Pool Storage Fraction 2

Type: Periodic Slot
Units: Fraction
Description: The values in the periodic slot represent the percentage of the conservation pool storage at each level (column) in the Operating Level Storage 2 Table.
Information: It has identical dimension including dates and levels as the Operating Level Storage 2 Table. This table will be populated at beginning of run. The Operating Level 2 Table will be its “source” slot.
I/O: Output only
Links: Not Linkable

☛ Flood Pool Storage Fraction 2

Type: Periodic Slot
Units: Fraction
Description: The values in the periodic slot represent the percentage of the flood pool storage at each level (column) in the Operating Level 2 Table.
Information: It has identical dimension including dates and levels as the Operating Level Storage 2 Table. This table will be populated at beginning of run. The Operating Level 2 Table will be its “source” slot.
I/O: Output only
Links: Not Linkable

Method Details:

This method category will be dependent on the selection of the **Conservation and Flood Pools** or **Conservation Pool** method in the **Operating Levels** method category.

At the beginning of the run, the Operating Level Storage 2 Table will be populated by looking up the elevation values in the Operating Level 2 Table on the Elevation Volume Table to get the storage associated with each level. Next the Conservation Pool Storage Fraction 2 and Flood Pool Storage Fraction 2 slots will be populated as follows:

For each date (row) and each level, n, (column), the equation to compute the fraction:

$$\text{Conservation Pool Storage Fraction 2}[date,n] =$$

$$\frac{\text{Op Level Storage 2 Table}[t, n] - \text{Op Level Storage 2 Table}[t, \text{Bottom of Cons Pool}]}{\text{Op Level Storage 2 Table}[t, \text{Top of Cons Pool}] - \text{Op Level Storage 2 Table}[t, \text{Bottom of Cons Pool}]}$$

Note, the **Conservation Pool Storage Fraction 2** is limited to be always between 0 and 1 (0% to 100%).

For each date (row) and each level, n, (column), the equation to compute the fraction:

$$\text{Flood Pool Storage Fraction}[date,n] =$$

$$\frac{\text{Op Level Storage 2 Table}[t, n] - \text{Op Level Storage 2 Table}[t, \text{Top of Cons Pool}]}{\text{Op Level Storage 2 Table}[t, \text{Top of Flood Pool}] - \text{Op Level Storage 2 Table}[t, \text{Top of Cons Pool}]}$$

Note, the Flood Pool Storage Fraction 2 is not limited to be between 0 and 1 (0% to 100%). For levels above the flood pool, the percentage will be greater than 100%.

Then, at the beginning of each timestep, the Sum Inflows over Interval method will check to see if the controller is on the “Trigger DateTime”. If so, the Inflow Sum slot will be summed (as a volume) over the criteria interval. If the sum is equal to or greater than the Operating Level 2 Trigger Volume, the reservoir will use the Operating Level 2 Table and Operating Level Storage 2 Table in all computations (until reset).

If the current timestep is a “Reset DateTime”, then the reservoir will again use the original Operating Level Table(s).

If the current timestep is neither a “Trigger DateTime” or a “Reset DateTime”, then the reservoir will reference the table used on the previous timestep. That is, it will not modify the table used but continue to use whichever table is in effect.

21.1.32 Startup

This category depends on selecting the Unit Power Table method, and describes how the monetary cost associated with starting up or shutting down a unit (turbine) will be modeled. There are two methods in this category, one which does not model these costs (effectively assigning them a value of 0) and one which uses a table describing the combined costs for starting up and shutting down a unit.

21.1.32.1 No Method (default)

This is the default, do-nothing method.

21.1.32.2 Unit Lumped Cost Method

For each Unit, this method lumps the cost of startup and shutdown into one value.

SLOTS ADDED BY THIS METHOD

Note, many of these slots have column or row dimensions based on the number of units. The rows/columns of these slots are expanded at the beginning of the run to match the value in the Number of Units slot. When first configuring this method, the user must enter the Number of Units, then run the model (stepping through 1 timestep is enough) to grow the slots to the right dimensions.

The following slots are instantiated when this method is selected:

☛ **UNIT STARTUP COST TABLE**

Type: TableSlot
Units: VALUE ($)
Description: This table will indicate the cost of startup/shutdown of each unit.
Information: There will be one column for each unit and one row that represents the cost of startup/shutdown.
I/O: Required input
Links: NA

☛ **UNIT STARTUP COST**

Type: AggSeriesSlot
Units: VALUE ($)
Description: There is one column for each unit indicating the cost of startup/shutdown.
Information: In simulation, the value of Unit Startup Cost for each unit is the Unit Startup[u] * Unit Startup Cost Table [u].
I/O: Output only
Links: NA

☛ Unit Startup

Type: AggSeriesSlot
Units: No Units
Description: A value of 1 indicates that the unit starts up at that date; otherwise the value is 0, indicating that the unit does not start up at that date.
Information: There is one column for each unit.
I/O: Output only
Links: NA

☛ Unit Shutdown

Type: AggSeriesSlot
Units: No Units
Description: A value of 1 indicates that the unit shuts down at that date; otherwise the value is 0, indicating that the unit does not shut down at that date.
Information: There is one column for each unit.
I/O: Output only
Links: NA

☛ Number Of Units Startup

Type: SeriesSlot
Units: No Units
Description: The number of units which start up at a given date. This value is the sum over the columns of Unit Startup.
Information:
I/O: Output only
Links: NA

☛ Number Of Units Shutdown

Type: SeriesSlot
Units: No Units
Description: The number of units which shut down at a given date. This value is the sum over the columns of Unit Shutdown.
Information:
I/O: Output only
Links: NA

☛ Plant Startup Cost

Type: SeriesSlot
Units: Value ($)
Description: The total startup cost for the plant. This value is the sum over the columns of the Unit Startup Cost.
Information:

I/O: Output only

Links: Linkable, typically to a Thermal object's System Startup slot.

Method Details

In Simulation, if the Unit Lumped Cost method is selected, startup and shutdown will be summarized as follows:

- Calculate Unit Startup[t,u] = max(Unit Is Generating[t,u] - Unit Is Generating[t-1,u], 0)
- Calculate Unit Shutdown[t,u] = max(Unit Is Generating[t-1,u] - Unit Is Generating[t,u], 0)
- Calculate Number Of Units Startup[t] = sum(Unit Startup[t])
- Calculate Number Of Unit Shutdown[t] = sum(Unit Shutdown[t])

Note, if the previous Unit Is Generating is not known, it is assumed that the unit is neither starting up or shutting down; Unit Startup and Unit Shutdown are set to zero. This may happens on the start timestep when the previous value is not known. Also, if the current Unit Is Generating is not valid, the method is exited without performing any computations or setting any slots.

This method will calculate the cost associated with startup/shutdown for each unit and the plant:

- Unit Startup Cost[t,u] = Unit Startup[t,u] * Unit Startup Cost Table[u]
- Plant Startup Cost[t] = Σ (Unit Startup Cost[t])

21.1.33 Head Loss

This category depends on the Unit Power Table method and contains methods for modeling additional head loss that occurs. This head loss may come from the configuration of the penstocks for bringing water to the turbines.

21.1.33.1 No Method (default)

In this method, there is no additional head loss to be used in the power calculation. In terms of penstock head loss, this method should be selected if the penstocks for the units are independent and the penstock losses are typically incorporated in the power data. Thus the power data is specified in terms of operating head.

21.1.33.2 Shared Penstock Head Loss method

In this method, there is additional head loss that results because units share a common penstock. The operating head losses in the penstock depend on the total turbine release and are shared for all units. The net head is calculated by subtracting penstock losses from the operating head. The unit data and power must be specified in terms of unit Net Heads instead of Operating Head.

Slots Added by this Method

The following slots are instantiated when this method is selected:

☛ **Shared Penstock Head Loss Table**

Type: TableSlot
Units: Flow vs Length
Description: This table shows head losses in a shared penstock as a function of total turbine release.
Information: The table has two columns: Turbine Release and Shared Penstock Loss.
I/O: Required Input
Links: NA

Method Details

In simulation, when either the Unit Power Table or Unit Power Table Release method is called, it has an estimate of operating head calculated as average PE minus average tailwater elevation. Within these methods, if the Shared Penstock Head Loss method is selected, the code will look up an estimate of total Turbine Release and compute the additional head loss and subtract it from the existing operating head before use in any other equation. An iterative solution is needed if Turbine Release is not known, i.e. if this method is called from the Unit Power Table Release method.

21.1.34 Cavitation

This category depends on selecting the Unit Power Table method and contains methods for dealing with the problem of cavitation on turbines. Cavitation is the sudden formation and collapse of low-pressure bubbles in liquids by means of mechanical forces and this process can cause damage to turbines under certain operating conditions.

21.1.34.1 No Method (default)

This is the default, do-nothing method.

21.1.34.2 Unit Head and Tailwater Based Regions

This method allows the user to specify the regions of operation in which cavitation does NOT occurs, so that these regions can be avoided. These regions can be dependent on both operating head and tailwater.

Slots added by this method

Note, many of these slots have column or row dimensions based on the number of units. The rows/columns of these slots are expanded at the beginning of the run to match the value in the Number of Units slot. When first configuring this method, the user must enter the Number of Units, then run the model (stepping through 1 timestep is enough) to grow the slots to the right dimensions.

The following slots are instantiated when this method is selected:

☛ **Unit Power Cavitation Table**

Type: TableSlot
Units: Length, Length, Power, and Power
Description: This table represents the region of operation that does not cause cavitations.
Information: The table will have one block per unit and four columns per block: head, tailwater, and minimum power to prevent cavitation, and maximum power to prevent cavitation. Interpolation of this table will be used to determine the feasibility for each flow - head combinations in the Unit Power Table. Some combinations may not be feasible at any tailwater, and these combinations should not be used in optimization or simulation. Others will have a minimum tailwater level for feasibility. For some units, tailwater may not affect cavitation. In these cases two rows should be used for each head: one with minimum tailwater and one with maximum tailwater.
I/O: Required Input
Links: NA

☛ **Unit Cavitation Optimization Tolerances**

Type: Table

Unit 1				Unit 2			
Tailwater (ft)	Operating Head (ft)	Min. Non-Cav. Power (MW)	Max. non-Cav. Power (MW)	Tailwater (ft)	Operating Head (ft)	Min. Non-Cav. Power (MW)	Max. non-Cav. Power (MW)
2067	100	1	12	2067	100	2	10
2067	200	1.1	12.5	2067	200	3	11
2067	300	1.4	12.6	2067	300	3.5	12
2116	100	2	12	2116	100	2	10
2116	200	2.1	12.3	2116	200	3	11
2116	300	2.5	12.8	2116	300	3.5	12

Units: FRACTION, FRACTION
Description: Tolerance used to adjust the cavitation region in Optimization.
Information: This slot is used to "shrink" the cavitation region in Optimization to avoid the possibility of optimal solutions that when run in the Rulebased Simulation, just barely dip into a cavitation zone. For example, if 0.01 is specfied, this translates to giving the optimization a 1% cushion to avoid the cavitation zone. There is one row for each unit.
I/O: Optional Input
Links: NA

METHOD DETAILS

In Simulation, the dispatch method will execute this method once head, tailwater elevation, and power are computed for each unit. This method will determine if the computed head, tailwater elevation, and power fall outside of the minimum and maximum power to prevent cavitation regions. If so, the method will issue an error but not stop the run. If the method is called and there is no valid Unit Power, then the method is exited without performing any computations.

21.1.35 Avoidance Zones

This category depends on selecting the Unit Power Table method and contains methods for modeling the existence of undesirable regions of operation for turbines. There are two methods in this category, one which does not model avoidance zones at all, and one which

21.1.35.1 No Method (default)

This the default, do-nothing method; avoidance zones are not considered.

21.1.35.2 Unit Head Based Avoidance Zones

This method allows the user to specify a table that defines the conditions in which the turbines should not be operated.

Slots added by this method

Note, many of these slots have column or row dimensions based on the number of units. The rows/columns of these slots are expanded at the beginning of the run to match the value in the Number of Units slot. When first configuring this method, the user must enter the Number of Units, then run the model (stepping through 1 timestep is enough) to grow the slots to the right dimensions.

The following slots are instantiated when this method is selected:

☛ **Unit Avoidance Zones Table**

Type: TableSlot

Units: Length, Power, Power

Description: This table represents zones in the Unit Power Table that should be avoided for operations.

Information: The avoidance zone table has one block per unit and three columns per block: head, power at bottom of the avoidance zone, power at the top of the avoidance zone. This table effectively removes regions from the Unit Power Table. The regions removed might have to be interpolated between points in the table. Heads that appear in this table must appear in the Unit Power Table as well.

I/O: Required Input

Links: NA

Unit 1			Unit 2		
Operating Head (ft)	Min. Power at Zone Bottom (MW)	Max. Power at Zone Top (MW)	Operating Head (ft)	Min. Power at Zone Bottom (MW)	Max. Power at Zone Top (MW)
100	4.5	6	100	8	9
200	5	6.4	200	9	9.5
300	6	7	300	9.2	9.9

Method Details

In Simulation, the dispatch method will execute this method once head and power are computed for each unit. This method will determine if the computed head and power fall inside an avoidance zone. If so, the method will issue an error but not stop the run. If the method is called and there is no valid Unit Power, then the method is exited without performing any computations.

21.1.36 Frequency Regulation

This category depends on selecting the Unit Power Table method, although in the future it might be enabled for other power methods. The frequency regulation methods model the provision of the frequency regulation ancillary service, that is, how the reservoir can be made available to flexibly follow a load demand within a specified range during a certain period in order to affect the frequency of the generated power.

21.1.36.1 No Method (default)

This is the default, do-nothing method; no regulation is modeled.

21.1.36.2 Unit Frequency Regulation

NOTE: ALTHOUGH THE METHOD CAN BE SELECTED AND SLOTS ARE ADDED, THIS METHOD IS NOT YET IMPLEMENTED.

When frequency regulation is scheduled, it allows the unit to follow the real time load. Exactly what will happen in real time is unknowable. This results in two sets of values at scheduling time, nominally scheduled power and turbine release. It is uncertain if the real time operators will actually use the service. At present, we distinguish between the nominal "scheduled" power (and turbine release) that the regulation is allowed to depart from and the "expected" power generation (and turbine release) that will take place when regulation is allowed. Both are important. The scheduled power sets the baseline for regulation and should be communicated to the power dispatchers. The expected power and release are more useful for coordinating a plant with the rest of the system.

SLOTS ADDED BY THIS METHOD

Note, many of these slots have column or row dimensions based on the number of units. The rows/columns of these slots are expanded at the beginning of the run to match the value in the Number of Units slot. When first configuring this method, the user must enter the Number of Units, then run the model (stepping through 1 timestep is enough) to grow the slots to the right dimensions.

The following slots are instantiated when this method is selected:

☛ **UNIT REGULATION TABLE**

Type: TableSlot

Units: LENGTH, FLOW, POWER, FLOW, POWER

Description: This table (not visible to the user) represents the available regulation (both up and down in terms of flow and power) for each unit at each point in the Unit Power Table.

Information: This table is calculated using data in the Unit Power Table and the Avoidance Zone Table (if applicable) and could be calculated automatically at beginning of run in simulation and/or optimization. This table consists of a block of six

columns for each unit. The head and flow values should be the same as the Unit Power Table. The other four columns in the block are respectively Regulation Flow Up, Regulation Power Up, Regulation Flow Down, and Regulation Power Down. These values represent the minimum and maximum power achievable from the initial flow value without passing through an avoidance zone.We require that the heads in this table appear in the Unit Power Table as well.

I/O: Automatically calculated at beginning of run

Links: NA

Unit Two Sided Regulation

Type: AggSeriesSlot
Units: Power
Description: The value is the two sided frequency regulation for the unit at that timestep.
Information: There is one column for each unit.
I/O: Input or Output

Links: NA

Unit Regulation Up

Type: AggSeriesSlot
Units: Power
Description: The value is the frequency regulation up for the unit at that timestep.
Information: There is one column for each unit.
I/O: Input or Output

Links: NA

Unit Regulation Down

Type: AggSeriesSlot
Units: Power
Description: The value is the frequency regulation down for the unit at that timestep.
Information: There is one column for each unit.
I/O: Input or Output

Links: NA

Unit Possible Regulation Up

Type: AggSeriesSlot
Units: Power
Description: The value is the possible regulation up for the unit at that timestep.
Information: There is one column for each unit.
I/O: Input or Output

Links: NA

☛ Unit Possible Regulation Down

Type: AggSeriesSlot
Units: Power
Description: The value is the possible regulation down for the unit at that timestep.
Information: There is one column for each unit.
I/O: Input or Output
Links: NA

☛ Unit Flow Addition For Regulation

Type: AggSeriesSlot
Units: Flow
Description: The value is the additional release required to reach the frequency high point for the unit at that timestep.
Information: There is one column for each unit. This value is typically returned from optimization and set via a rule.
I/O: Rule
Links: NA

☛ Unit Flow Reduction For Regulation

Type: AggSeriesSlot
Units: Flow
Description: The value is the reduction in release required to reach the frequency low point for the unit at that timestep.
Information: There is one column for each unit. This value is typically returned from optimization and set via a rule.
I/O: Rule
Links: NA

☛ Unit Scheduled Mechanical Power

Type: AggSeriesSlot
Units: Power
Description: The value is the scheduled mechanical power generation, before subtracting regulation (or reactive power) for the unit at that timestep.
Information: There is one column for each unit.
I/O: Output only
Links: NA

☛ Unit Scheduled Turbine Release

Type: AggSeriesSlot
Units: Flow
Description: The value is the turbine flow which corresponds to the Unit Scheduled mechanical Power for the unit at that timestep.

Information: There is one column for each unit.
I/O: Output only
Links: NA

☛ Unit Operating Cost Per Regulation Table
Type: TableSlot
Units: VALUE ($)
Description: For each generating unit, this is the cost per unit of regulation.
Information: There is one row for each unit.
I/O: Input
Links: NA

☛ Unit Operating Cost
Type: AggSeriesSlot
Units: VALUE ($)
Description: This is the total cost of using a unit for regulation incurred during the run.
Information: There is one column for each unit.
I/O: Output
Links: NA

☛ Operating Cost
Type: AggSeriesSlot
Units: VALUE ($)
Description: The value is the sum of the unit operating costs.
Information: This is an existing slot with only one column.
I/O: Output only
Links: NA

☛ Regulation
Type: SeriesSlot
Units: POWER
Description: Total regulation for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Two Sided Regulation.
I/O: Output only
Links: Optional

☛ Plant Regulation Up
Type: SeriesSlot
Units: POWER
Description: Total regulation up for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Regulation Up.
I/O: Input or Output

Links: Linkable, typically to the Thermal object System Regulation Up

☛ Plant Regulation Down

Type: SeriesSlot
Units: POWER
Description: Total regulation down for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Regulation Down.
I/O: Input or Output
Links: Linkable, typically to the Thermal object System Regulation Down

☛ Plant Possible Regulation Up

Type: SeriesSlot
Units: POWER
Description: Total possible regulation up for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Possible Regulation Up.
I/O: Output only
Links: No

☛ Plant Possible Regulation Down

Type: SeriesSlot
Units: POWER
Description: Total possible regulation down for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Possible Regulation Down.
I/O: Output only
Links: Optional

☛ Plant Flow Addition For Regulation

Type: SeriesSlot
Units: FLOW
Description: Total additional turbine release required in order to reach the frequency regulation high point for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Flow Addition For Regulation.
I/O: Output only
Links: Optional

☛ Plant Flow Reduction For Regulation

Type: SeriesSlot
Units: FLOW
Description: Total reduction in turbine release required in order to reach the frequency regulation low point for the reservoir (plant) at that timestep.

Information: This value is the sum over the columns of Unit Flow Reduction For Regulation.
I/O: Output only
Links: No

PLANT SCHEDULED MECHANICAL POWER

Type: SeriesSlot
Units: POWER
Description: Total scheduled mechanical power for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Scheduled Mechanical Power.
I/O: Output only
Links: No

PLANT SCHEDULED TURBINE RELEASE

Type: SeriesSlot
Units: FLOW
Description: Total scheduled turbine release for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Scheduled Turbine Release.
I/O: Output only
Links: No

In Simulation, the Unit Power Table method will execute this method when the Unit Turbine Release is known. At this time, this method cannot be called if Unit Energy is specified (input or rules):

- Unit Scheduled Turbine Release[t,u] = Unit Turbine Release[t,u] - (Unit Flow Addition for Regulation[t,u] + Unit Flow Reduction For Regulation [t,u])/2
- Unit Scheduled Mechanical Power[t,u] = Unit Power Table(head[t], Unit Scheduled Turbine Release[t,u])
- Unit Regulation Up[t.u] = Unit Power Table(head[t], Unit Scheduled Turbine Release[t,u] + Unit Flow Addition for Regulation[t,u] /2)
- Unit Regulation Down [t,u] = Unit Power Table(head[t], Unit Scheduled Turbine Release[t,u]-Unit Flow Reduction For Regulation [t,u]/2)
- Unit Regulation [t,u] = max(Unit Regulation Up[t.u] , Unit Regulation Down [t,u])
- Unit Power[t,u] = Unit Scheduled Mechanical Power[t,u] + Unit Regulation Up[t.u]/2 - Unit Regulation Down [t,u]/2
- Calculate Unit Operating Cost[t,u] = timestep * Unit Operating Cost Per Regulation[u] * (Unit Regulation Up[t,u] + Unit Regulation Down[t,u])

- Compute plant level values as a sum of unit values

21.1.37

21.2 Dispatch Methods

Unless otherwise specified, all the dispatch methods listed below are available when the Pump and Release Only method is selected in the Pump and Release Accommodation category.

21.2.1 solveMB_givenPumpsUsedInflow

Solves the mass balance equation when the number of pumping units is given.

REQUIRED KNOWNS

- **DIVERSION**
- **RETURN FLOW**
- **TAILWATER BASE VALUE**
- **PUMPS USED**
- **INFLOW**

REQUIRED UNKNOWNS

☛ Pumped Flow

☛ Pump Power

☛ Storage

☛ Pool Elevation

☛ Outflow

☛ Energy

The method first checks if the Canal object is linked. If the Canal has not yet solved, the dispatch method waits for it to solve before continuing. Then, the pump power calculation method is executed. Inflow, Outflow, Pump Power and Energy are solved within the pump power calculations.

Storage is solved by the mass balance equation:

$$Storage = Storage(t-1) + Inflow - Outflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods. The inflow term includes the effects of hydrologic inflow, diversion and return flow. Since the evaporation, bank storage, seepage and precipitation terms depend upon the Storage value, the mass balance solution is an iterative process. (See note below.) Pool Elevation is calculated using the Elevation Volume Table and the calculated Storage value.

Gate setting, energy in storage, spilled energy, and future value calculations are then performed if the user has selected them. The dispatch method is then complete.

NOTE: In specific cases when Storage is very close to zero, the iterative mass balance solution will follow one of two possible processes. The first process allows the loop to iterate in the negative storage range before final convergence. Convergence in this situation is typically quite rapid, and in instances when the final storage is, in fact, negative, an error is posted stating that the outflow is too large to be physically possible. This process is invoked if an additional row is appended to the Elevation Volume table specifying a negative storage value within which the loop can iterate. If specifying and allowing negative storage values in the iteration is not desired, no negative storage value should be appended to the Elevation Volume table. In this second process, the algorithm uses storage equals zero whenever it is in the negative storage range. If the outflow is really too great to be physically possible, the algorithm will keep iterating until it reaches maximum iterations. If this happens, RiverWare

does a final mass balance check at the storage equals zero point and posts an error stating that the outflow is too large.

21.2.2 solveMB_givenPumpedFlowInflow

Solves the mass balance equation when Inflow is known.

Required Knowns

- Diversion
- Return Flow
- Inflow
- Pumped Flow

Required Unknowns

- Pumps Used
- Pump Power
- Storage
- Pool Elevation
- Outflow
- Energy

The method first checks if the Canal object is linked. If the Canal has not yet solved, the dispatch method waits for it to solve before continuing. Then, the pump power calculation method is executed. Pumps Used, Outflow, Pump Power and Energy are solved within the pump power calculations.

Storage is solved by the mass balance equation:

$$Storage = Storage(t-1) + Inflow - Outflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods. The inflow term includes the effects of hydrologic inflow, diversion and return flow. Since the evaporation, bank storage, seepage and precipitation terms depend upon the Storage value, the mass balance solution is an iterative process. (See note below.) Pool Elevation is calculated using the Elevation Volume Table and the calculated Storage value.

Gate setting, energy in storage, spilled energy, and future value calculations are then performed if the user has selected them. The dispatch method is then complete.

NOTE: In specific cases when Storage is very close to zero, the iterative mass balance solution will follow one of two possible processes. The first process allows the loop to iterate in the negative storage range before final convergence. Convergence in this situation is typically quite rapid, and in instances when the final storage is, in fact, negative, an error is posted stating that the outflow is too large to be physically possible. This process is invoked if an additional row is appended to the Elevation Volume table specifying a negative storage value within which the loop can iterate. If specifying and allowing negative storage values in the iteration is not desired, no negative storage value should be appended to the Elevation Volume table. In this second process, the algorithm uses storage equals zero whenever it is in the negative storage range. If the outflow is really too great to be physically possible, the algorithm will keep iterating until it reaches maximum iterations. If this happens, RiverWare does a final mass balance check at the storage equals zero point and posts an error stating that the outflow is too large.

21.2.3 solveMB_givenPumpPowerInflow

Solves the mass balance equation when Pump Power is known.

Required Knowns

- ☛ **Diversion**
- ☛ **Return Flow**
- ☛ **Tailwater Base Value**
- ☛ **Pump Power**
- ☛ **Inflow**

Required Unknowns

☛ Pumps Used

☛ Pumped Flow

☛ Storage

☛ Pool Elevation

☛ Outflow

☛ Energy

The method first checks if the Canal object is linked. If the Canal has not yet solved, the dispatch method waits for it to solve before continuing. Then, the pump power calculation method is executed. Pumps Used, Outflow, Inflow and Energy are solved within the pump power calculations.

Storage is solved by the mass balance equation:

$$Storage = Storage(t-1) + Inflow - Outflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods. The inflow term includes the effects of hydrologic inflow, diversion and return flow. Since the evaporation, bank storage, seepage and precipitation terms depend upon the Storage value, the mass balance solution is an iterative process. (See note below.) Pool Elevation is calculated using the Elevation Volume Table and the calculated Storage value.

Gate setting, energy in storage, spilled energy, and future value calculations are then performed if the user has selected them. The dispatch method is then complete.

NOTE: In specific cases when Storage is very close to zero, the iterative mass balance solution will follow one of two possible processes. The first process allows the loop to iterate in the negative storage range before final convergence. Convergence in this situation is typically quite rapid, and in instances when the final storage is, in fact, negative, an error is posted stating that the outflow is too large to be physically possible. This process is invoked if an additional row is appended to the Elevation Volume table specifying a negative storage value within which the loop can iterate. If specifying and allowing negative storage values in the iteration is not desired, no negative storage value should be appended to the Elevation Volume table. In this second process, the algorithm uses storage equals zero whenever it is in the negative storage range. If the outflow is really too great to be physically possible, the algorithm will keep iterating until it reaches maximum iterations. If this happens, RiverWare

does a final mass balance check at the storage equals zero point and posts an error stating that the outflow is too large.

21.2.4 solveMB_givenInflowHW

This dispatch routine determines if the reservoir is draining or filling, and then refers to the appropriate dispatch method for the reservoir condition.

Required Knowns

- ☛ Diversion
- ☛ Return Flow
- ☛ Pool Elevation
- ☛ Inflow

Required Unknowns

- ☛ Pumped Flow
- ☛ Pump Power
- ☛ Pumps Used
- ☛ Storage
- ☛ Outflow
- ☛ Energy

The method first checks if the Canal object is linked. If the Canal is linked and has not yet solved, the dispatch method waits for it to solve before continuing. If Pool Elevation is flagged TARGET, an error is posted. Target operations are not valid for the Pumped Storage Reservoir. Storage is then calculated using the Elevation Volume Table and the known Pool Elevation.

Once the checks have passed, Outflow is set equal to zero and Inflow is calculated by the mass balance equation:

$$Inflow = Storage - Storage(t-1) + Outflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods.

If the calculated value for Inflow is greater than zero, the reservoir is filling and the solveMB_givenInflow method is executed. If Inflow is less than zero, the reservoir is releasing and Outflow is set as the absolute value of Inflow. The solveMB_givenOutflow method is then executed.

21.2.5 solveMB_givenInflowStorage

This dispatch routine determines if the reservoir is draining or filling, and then refers to the appropriate dispatch method for the reservoir condition.

Required Knowns

- ☛ **Diversion**
- ☛ **Return Flow**
- ☛ **Storage**
- ☛ **Inflow**

Required Unknowns

- ☛ **Pumped Flow**
- ☛ **Pump Power**
- ☛ **Pumps Used**
- ☛ **Pool Elevation**
- ☛ **Outflow**
- ☛ **Energy**

The method first checks if the Canal object is linked. If the Canal is linked and has not yet solved, the dispatch method waits for it to solve before continuing. If Storage is flagged

TARGET, an error is posted. Target operations are not valid for the Pumped Storage Reservoir. Pool Elevation is then calculated using the Elevation Volume Table and the known Storage value.

Once the checks have passed, Outflow is set equal to zero and Inflow is calculated by the mass balance equation:

$$Inflow = Storage - Storage(t-1) + Outflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods.

If the calculated value for Inflow is greater than zero, the reservoir is filling and the solveMB_givenInflow method is executed. If Inflow is less than zero, the reservoir is releasing and Outflow is set as the absolute value of Inflow. The solveMB_givenOutflow method is then executed.

21.2.6 solveMB_givenOutflowInflow

Solves the mass balance equation when Outflow is known. It is executed when the following list of knowns and unknowns are met or when it is called from either solveMB_givenHW or solveMB_givenStorage. Pump power calculations are not utilized in this method because it assumes the reservoir is only releasing water.

Required Knowns

- ☛ **Diversion**
- ☛ **Return Flow**
- ☛ **Outflow**
- ☛ **Inflow**

Required Unknowns

☛ Pumped Flow

☛ Pump Power

☛ Pump Energy

☛ Pumps Used

☛ Pool Elevation

☛ Energy

☛ Storage

The method first checks if the Canal object is linked. If the Canal is linked and has not yet solved, the dispatch method waits for it to solve before continuing. If Outflow is flagged MAX CAPACITY, it is set as the sum of the maximum spill and maximum turbine release. If this dispatch method is not being called from solveMB_givenHW or solveMB_givenStorage, Storage and Pool Elevation must be calculated from mass balance (inflow is set to zero):

$$Storage = Storage(t-1) + Inflow - Outflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods. The inflow term includes the effects of hydrologic inflow, diversion and return flow. Since the evaporation, bank storage, seepage and precipitation terms depend upon the Storage value, the mass balance solution is an iterative process. (See note below.) Pool Elevation is calculated using the Elevation Volume Table and the calculated Storage value.

If the dispatch method is being called from solveMB_givenHW or solveMB_givenStorage, Storage and Pool elevation are already known.

The spill method is then executed followed by the power calculation method. After Turbine Release has been calculated, the spill method may be executed again if there are excess outflows.

Gate setting, energy in storage, spilled energy and future value calculations are performed if the user has selected them. The dispatch method is then complete.

NOTE: In specific cases when Storage is very close to zero, the iterative mass balance solution will follow one of two possible processes. The first process allows the loop to iterate in the negative storage range before final convergence. Convergence in this situation is typically quite rapid, and in instances when the final storage is, in fact, negative, an error is posted stating that the outflow is too large to be physically possible. This process is invoked if an additional row is appended to the Elevation Volume table specifying a negative storage value within which the loop can iterate. If specifying and allowing negative storage values in the iteration is not desired, no negative storage value should be appended to the Elevation Volume table. In this second process, the algorithm uses storage equals zero whenever it is in the negative storage range. If the outflow is really too great to be physically possible, the algorithm will keep iterating until it reaches maximum iterations. If this happens, RiverWare does a final mass balance check at the storage equals zero point and posts an error stating that the outflow is too large.

21.2.7 solveMB_givenEnergyInflow

Solves the mass balance equation when Energy is known. It assumes that there is outflow through the turbine but no inflow through the pumps.

Required Knowns

- ☛ **Diversion**
- ☛ **Return Flow**
- ☛ **Tailwater Base Value**
- ☛ **Energy**
- ☛ **Inflow**

Required Unknowns

☞ Pumped Flow

☞ Pump Power

☞ Pump Energy

☞ Pumps Used

☞ Pool Elevation

☞ Outflow

☞ Storage

The method first checks if the Canal object is linked. If the Canal object is linked but has not yet solved, the dispatch method waits for it to solve before continuing.

Inflow is set to zero because there is no flow entering through the pumps. Outflow and Storage must then be solved by iteration. First, the spill, tailwater, and power release calculation methods are executed. Then Outflow is calculated as the sum of Turbine Release and Spill. With the new value of Outflow, Storage is solved as follows:

$$Storage = Storage(t-1) + Inflow - Outflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods. The inflow term includes the effects of hydrologic inflow, diversion and return flow. Since the evaporation, bank storage, seepage and precipitation terms depend upon the Storage value, the mass balance solution is an iterative process. (See note below.) Pool Elevation is calculated using the Elevation Volume Table and the calculated Storage value. The process is repeated with the new Storage value until the solution converges.

Power, gate setting, energy in storage, spilled energy, and future value calculations are performed if the user has selected them. The dispatch method is then complete.

NOTE: In specific cases when Storage is very close to zero, the iterative mass balance solution will follow one of two possible processes. The first process allows the loop to iterate in the negative storage range before final convergence. Convergence in this situation is typically quite rapid, and in instances when the final storage is, in fact, negative, an error is posted stating that the outflow is too large to be physically possible. This process is invoked if an additional row is appended to the Elevation Volume table specifying a negative storage value within which the loop can iterate. If specifying and allowing negative storage values in the iteration is not desired, no negative storage value should be appended to the Elevation

Volume table. In this second process, the algorithm uses storage equals zero whenever it is in the negative storage range. If the outflow is really too great to be physically possible, the algorithm will keep iterating until it reaches maximum iterations. If this happens, RiverWare does a final mass balance check at the storage equals zero point and posts an error stating that the outflow is too large.

21.2.8 solveMB_givenPumpedFlowOutflowInflow

This method is only available when Pump and Release is selected in the Pump and Release Accommodation category. It solves the object when Pumped Flow, Outflow, and Inflow are known.

Required Knowns

- Diversion
- Return Flow
- Inflow
- Outflow
- Pumped Flow

Required Unknowns

- Pumps Used
- Pump Power
- Pump Energy
- Storage
- Pool Elevation
- Energy

This dispatch method checks to see if either Outflow or Pumped Flow is zero.

- If neither are zero, an error is flagged and the run is aborted because a non-zero number cannot be given to both Pumped Flow and Outflow.
- If Outflow is zero, then the reservoir is pumping and the solveMB_givenPumpedFlowInflow method is executed.
- If Pumped Flow is zero, then the reservoir is releasing and the solveMB_givenOutflowInflow method is executed.

21.2.9 solveMB_givenInflowPumpAndGenEnergy

This method is available for both the Pump or Release Only, and the Pump and Release methods in the Pump and Release Accommodation category. It is used when pumping and turbine release both occur in the same timestep.

Required Knowns

☛ **Diversion**

☛ **Return Flow**

☛ **Tailwater Base Value**

☛ **Energy**

☛ **Pump Energy**

☛ **Inflow**

Required Unknowns

☛ Pumped Flow

☛ Pumps Used

☛ Pool Elevation

☛ Outflow

☛ Storage

The method first checks if the Canal object is linked. If the Canal object is linked but has not yet solved, the dispatch method waits for it to solve before continuing.

An iterative procedure is used to solve for the unknowns. First, the Spill and Tailwater methods are executed. Then calculations are performed, similar to the Unit Pump Power method, which determine the Inflow and Pumps Used necessary to meet the given Pump Energy. The power release method is executed and Outflow is calculated as the sum of Turbine Release and Spill. With the new value of Outflow, Storage is solved as follows:

$$Storage = Storage(t-1) + Inflow - Outflow$$

The mass balance equation may include the effects of evaporation, bank storage, seepage, or precipitation depending on the selected user methods. The inflow term includes the effects of hydrologic inflow, diversion and return flow. Since the evaporation, bank storage, seepage and precipitation terms depend upon the Storage value, the mass balance solution is an iterative process. (See note below.) Pool Elevation is calculated using the Elevation Volume Table and the calculated Storage value. The process is repeated with the new Storage value until the solution converges.

Power, gate setting, energy in storage, spilled energy, and future value calculations are performed if the user has selected them. The dispatch method is then complete.

NOTE: In specific cases when Storage is very close to zero, the iterative mass balance solution will follow one of two possible processes. The first process allows the loop to iterate in the negative storage range before final convergence. Convergence in this situation is typically quite rapid, and in instances when the final storage is, in fact, negative, an error is posted stating that the outflow is too large to be physically possible. This process is invoked if an additional row is appended to the Elevation Volume table specifying a negative storage value within which the loop can iterate. If specifying and allowing negative storage values in the iteration is not desired, no negative storage value should be appended to the Elevation Volume table. In this second process, the algorithm uses storage equals zero whenever it is in the negative storage range. If the outflow is really too great to be physically possible, the algorithm will keep iterating until it reaches maximum iterations. If this happens, RiverWare

does a final mass balance check at the storage equals zero point and posts an error stating that the outflow is too large.

22. Reach

The Reach object models open channel flow. It contains User Methods to control how open channel flow is modeled. With these methods, the user can choose from very simple to very complex routing equations, and include or exclude the modeling of diversion, gains or losses, return flow, inflow, the management of stage and volume calculations. The reach can solve downstream in all cases, and it can solve upstream for some Routing methods.

General Slots

(slots which always appear for this object)

☛ INFLOW

Type: Series
Units: FLOW
Description: flow rate at entrance of the object
Information:
I/O: May be set as input or linked to another object.
Links: May be linked to the Outflow of any object.

☛ OUTFLOW

Type: Series
Units: FLOW
Description: flow rate at exit of the object
Information:
I/O: May be set as input (for certain Routing methods) or linked to another object.
Links: May be linked to the Inflow of any object.

22.1 User Methods

22.1.1 Routing

The Routing method determines how the flow is calculated. It can also effect what dispatch methods and user categories are available for use. Some of the Routing methods may only solve downstream (for outflow), while others may solve either upstream or downstream.

22.1.1.1 No Routing

No Routing is the default method for this category. This method involves a simple mass balance. It triggers the selection of dispatch methods which perform the actual calculations. It is the only method for which local Inflow may be calculated.

Slots Specific to this Method

☛ **Return Flow**

Type: Series
Units: FLOW
Description: return flow into the reach
Information: Enters at the bottom of the reach (i.e. is added directly to the outflow calculated by the method).
I/O: Optional; if not input and not linked, it is set to zero.
Links: May be linked to the Return Flow slot on any object, the Outflow slot on a Groundwater Storage object, the Excess GW Outflow slot on the Groundwater Storage object, or the Surface Return Flow slot on a WaterUser.

The calculations are executed by the dispatch methods solveNRInflow, solveNROutflow, and solveNRLocalInflow.

22.1.1.2 Time Lag

Inflows are lagged by a specified time to calculate outflows. This method can be solved upstream or downstream and therefore causes dispatching with either a known inflow or a known outflow. The lagtime can be any length of time.

Slots Specific to this Method

☛ LagTime

Type: Scalar
Units: TIME
Description: lagtime or travel time of a flow change through the reach
Information:
I/O: Required input
Links: Not linkable

A sample equation for Outflow is presented below:

$$Outflow((int)LagTime) = flowFrac \bullet Inflow + (1 - flowFrac) \bullet Inflow(-1) + flowFrac \bullet LocalInflow + (1 - flowFrac) \bullet LocalInflow(-1) + TotalGainLoss$$

where (-1) means the value at the previous timestep.

The equation for flow fraction:

$$flowFrac = 1 - \left(\frac{LagTime}{timestep} - (integerValueOf)\frac{LagTime}{timestep}\right)$$

This method can also solve for inflow, given outflow. Note that it is possible to set up a combination of input values for inflow and outflow which do not match. This will cause an error. An error will also result if the timestep is monthly and the Lag Time is not in months, or if the timestep is not monthly and the Lag Time is in months.

22.1.1.3 Variable Time Lag

Behaves similarly to Time Lag routing except that the Lag Time is allowed to vary as a function of the day of the year and the flow rate. This method may only solve downstream.

Slots Specific to this Method

☛ Return Flow

Type: Series
Units: FLOW
Description: return flow into the reach
Information: Enters at the bottom of the reach (i.e. is added directly to the outflow calculated by the method).
I/O: Optional; if not input and not linked, it is set to zero.
Links: May be linked to the Return Flow slot on any object, the Outflow slot on a Groundwater Storage object, the Excess GW Outflow slot on the Groundwater Storage object, or the Surface Return Flow slot on a WaterUser.

☛ Variable Lag Time

Type: Series
Units: TIME
Description: value of the Lag Time interpolated from the Variable LagTime Table.
Information:
I/O: Output only
Links: Not linkable

☛ Variable LagTime Table

Type: Table
Units: TIME VS. FLOW VS. TIME
Description: a table relating the day of the year, flow rate, and the lag time
Information: Data must be entered in blocks of increasing flow for each given day for the interpolation method to work correctly. January 1 is represented by a 1, February 1 as 32, etc. An example table is shown below.

Day of Year	Flow Rate	Lag Time
1	500	1.0
1	550	.9
1	600	.8
32	500	2.0
32	550	1.7
32	600	1.5
60	500	2.1
60	550	1.9
60	600	1.6

I/O: Required input
Links: Not linkable

This method executes when the Inflow is known for the current timestep. The Inflow value is then used to calculate the Variable Lag Time. Outflow values can then be solved for at the timesteps corresponding to the Variable Lag Time. A sample calculation is given below:

$$\text{Outflow(intger value of Variable Lag Time)} = \text{flowFrac1} \bullet (\text{Inflow} + \text{Local Inflow} - \text{Diversion})$$

$$\text{Outflow(integer value of Variable Lag Time + 1)} = \text{flowFrac2} \bullet (\text{Inflow} + \text{Local Inflow} - \text{Diversion})$$

where flowFrac1 and flowFrac2 to are calculated by the following equations:

$$\text{flowFrac1} = 1 - \left(\frac{\text{Variable Lag Time}}{\text{timestep}} - (integerValueOf)\frac{\text{Variable Lag Time}}{\text{timestep}}\right)$$

$$flowFrac2 = 1 - flowFrac1$$

In the equations given above, the integer value of a number means that the number is truncated after the decimal point with no rounding.

Since the lag time can vary with every timestep, it possible that more the one Inflow can contribute to a particular Outflow. Return Flow and Total Gain Loss may then be added to the calculated Outflow value.

22.1.1.4 Impulse Response

This function solves for the outflow from a reach given inflow.

SLOTS SPECIFIC TO THIS METHOD

☛ LAG COEFF

Type: Table
Units: NO UNITS
Description: impulse response coefficients
Information:
I/O: Required input
Links: Not linkable

☛ NUM. OF COEFF

Type: Scalar
Units: NO UNITS
Description: number of impulse response coefficients
Information:
I/O: Required input
Links: Not linkable

☛ RETURN FLOW

Type: Series
Units: FLOW
Description: return flow into the reach
Information: Enters at the bottom of the reach (i.e. is added directly to the outflow calculated by the method).
I/O: Optional; if not input and not linked, it is set to zero.
Links: May be linked to the Return Flow slot on any object, the Outflow slot on a Groundwater Storage object, the Excess GW Outflow slot on the Groundwater Storage object, or the Surface Return Flow slot on a WaterUser.

There must be the same number of values in the Lag Coeff table as the value given in Num of Coeff in order to successfully dispatch at any timestep. If any needed value is invalid, an error will occur, and the run will stop.

The general equation for this method is:

$$Outflow = C_0 Inflow_t + C_1 Inflow_{t-1} + \dots + C_{ncoeff-2} Inflow_{t-(ncoeff-2)} + C_{ncoeff-1} Inflow_{t-(ncoeff-1)} + TotalGainLoss$$

Note: If a new value is set for Inflow at a given timestep, the reach will re-dispatch to solve for a new Outflow at that timestep only. It will not, in general, re-solve for Outflow at every timestep that is affected by the new Inflow. For example, say the Outflow at timestep *t* is a function of the Inflow at *t, t* - 1, and *t* - 2. If a new Inflow value is set at timestep *t* - 2, the reach will redispatch and solve for a new Outflow at *t* - 2, but it will *not* re-dispatch at timesteps *t* - 1 and *t.* Thus the final Outflow values at *t* - 1 and *t* will not correspond to the updated Inflow value at *t* - 2. If this type of re-dispatching across multiple timesteps is required, the Step Response routing method should be used **HERE (Section 22.1.1.5)**.

22.1.1.5 Step Response

The step response method is a simple routing method which computes outflow for the current timestep and future timesteps given inflow values. The total number of outflows computed will be equal to the number of lag coefficients. The outflow will be computed as the sum of the routed inflows plus whatever sources or sinks may be available.

SLOTS SPECIFIC TO THIS METHOD

☛ RETURN FLOW

Type: Series
Units: FLOW
Description: return flow into the reach
Information: Enters at the bottom of the reach (i.e. is added directly to the outflow calculated by the method).
I/O: Optional; if not input and not linked, it is set to zero
Links: May be linked to the Return Flow slot on any object, the Outflow slot on a Groundwater Storage object, the Excess GW Outflow slot on the Groundwater Storage object, or the Surface Return Flow slot on a WaterUser.

☛ LAG COEFF

Type: Table
Units: NO UNITS
Description: step response coefficients
Information: The number of Lag Coefficients must be equal to the value in the Num. of Coeff slot.
I/O: Required input
Links: Not Linkable

☛ **NUM. OF COEFF**

Type: Scalar Slot
Units: NO UNITS
Description: number of step response coefficients
I/O: Required Input
Links: Not Linkable

The general calculations for this method are very similar to the Impulse Response routing method. First, the outflow for the current timestep will be calculated as:

$$Outflow = C_0 Inflow_t + C_1 Inflow_{t-1} + \ldots + C_{ncoeff-2} Inflow_{t-(ncoeff-2)} + C_{ncoeff-1} Inflow_{t-(ncoeff-1)} + TotalGainLoss$$

The method will then move on to the first future outflow. At this stage, outflow will be computed using the same equation only now, t will be incremented to t+1, t-1 incremented to t, etc. The total Gain Loss term will be the Total Gain Loss for timestep t+1.

In the situation where the method is looking for an inflow for a timestep that is actually past the current timestep and this value is not valid, this inflow will be assumed to be zero. If the inflow is not valid for a previous timestep, the calculation will exit and the method will move on to the next future timestep.

22.1.1.6 Variable Step Response

The Variable Step Response method is a variation of the Step Response routing method which computes outflow for the current timestep and future timesteps given inflow values. The total number of outflows computed will be equal to the number of lag coefficients. The outflow will be computed as the sum of the routed inflows plus whatever sources or sinks may be available. In the Step Response method, a single set of lag coefficients is used. In the Variable Step Response method, multiple sets of lag coefficients are used dependent on the inflows to the reach.

SLOTS SPECIFIC TO THIS METHOD

☛ **RETURN FLOW**

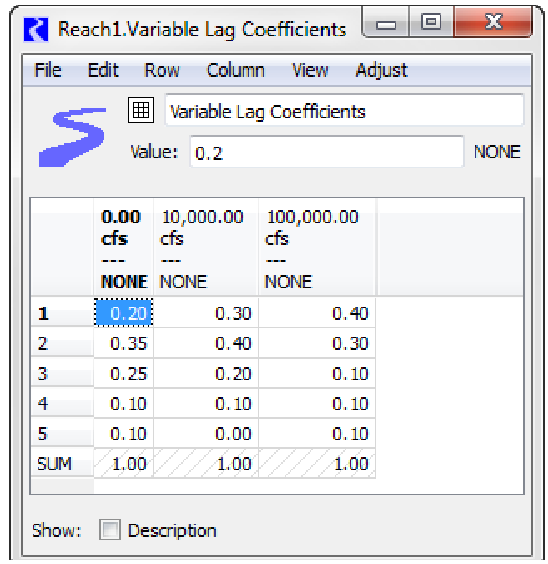
Type: Series
Units: FLOW
Description: return flow into the reach
Information: Enters at the bottom of the reach (i.e. is added directly to the outflow calculated by the method).
I/O: Optional; if not input and not linked, it is set to zero
Links: May be linked to the Return Flow slot on any object, the Outflow slot on a Groundwater Storage object, the Excess GW Outflow slot on the Groundwater Storage object, or the Surface Return Flow slot on a WaterUser.

☛ Variable Lag Coefficients

Type: Table Slot

Units: Column Map Values - Flow, Table values - No Units

Description: A table defining the step response coefficients for each inflow threshold as shown in the following sample.



Information: This table has a Column Map which means that each column has an associated numerical value (with units). This numerical value is displayed on as the column label. Columns are added and deleted from this table using the **Column** menu with the following options: **Set Number of Columns**, **Append Column**, **Delete Column** and **Delete Last Column**. User units, scale, type, and precision for the Column Map (i.e. the column heading values) are defined in the unit scheme for Flow unit types. Column map values are set from the **Column, Set Column Value** menu option. When a column value is changed, the columns will re-order to ensure that the column values are increasing left to right. The sum of coefficients in each column should equal 1.0. This can easily be verified by adding a summary row at the bottom of the table using the **View, Show Column Sum Row** menu option. When used in the Variable Step Response method, the column map values are used as a stair step lookup, e.g. an Inflow of 500cfs uses the first column, an inflow of 10,000cfs uses the second column, and an inflow of 120,000cfs uses the third column. Therefore, the column map values need not bound the highest expected flows; flows greater than the largest column map value use the right most column. Note, the left-most set of coefficients should represent the minimum flow in the reach.

I/O: Required Input

Links: Not Linkable

This Routing method is executed from the SolveOutflow dispatch method, i.e. it only solves for the Outflow given the Inflows, not vice versa. Note, Local Inflows are added to the top of the reach and are routed with the Inflows. In the following, Inflow refers to the sum of the Inflow and Local Inflow slot. The variable “ncoeff” is the number of rows in the Variable Lag Coefficients slot. The outflows will be calculated using the following algorithm:

For each t from (t = 0, i.e current timestep) to (t = current timestep + (ncoeff - 1))

$$Outflow_t = C_0 Inflow_t + C_1 Inflow_{t-1} + \\ \ldots + C_{ncoeff-2} Inflow_{t-(ncoeff-2)} + C_{ncoeff-1} Inflow_{t-(ncoeff-1)} + TotalGainLoss_t$$

End for

Each C coefficient is selected from the appropriate column of the Variable Lag Coefficients slot based on the value of the Inflow with which it is multiplied. For a particular evaluation of the equation, the sum of the coefficients does not necessarily equal 1.0. But, mass is preserved over time. Using the above sample table, if the Inflow at the current timestep t, is 12,000cfs and the Inflow at t-1 is 8,000cfs, C_0 is 0.30 and C_1 is 0.35 using the above table.

If the method is looking for an inflow for a timestep that is actually past the current timestep and the value is not valid, this inflow will be assumed to be zero. If the inflow is not valid for a previous timestep, the calculation will exit the method without calculating an outflow, post a warning message, and move on to the next timestep.

22.1.1.7 Muskingum

The Muskingum method is a simple storage routing method which solves for the propagation of flow waves based on a lag time and attenuation factor. Note that this method must have Inflow and Outflow known at the timestep before the first routing timestep.

The Muskingum method only allows for one routing segment, but allows for Local Inflows, Diversions, and Return Flow. The **Muskingum with Segments** method (described next) allows for multiple segments, but no Diversions, Local Inflows, or Return Flows.

Slots Specific to this Method

☛ Return Flow

Type: Series
Units: FLOW
Description: return flow into the reach
Information: Enters at the bottom of the reach (i.e. is added directly to the outflow calculated by the method).
I/O: Optional; if not input and not linked, it is set to zero.
Links: May be linked to the Return Flow slot on any object, the Outflow slot on a Groundwater Storage object, the Excess GW Outflow slot on the Groundwater Storage object, or the Surface Return Flow slot on a WaterUser.

The general equation for this method is:

$$Outflow = C_0 Inflow + C_1 Inflow(-1) + C_2 Outflow(-1) + TotalGainLoss$$

where outflow and outflow(-1) are the current and previous outflows respectively as with inflow and inflow(-1).

22.1.1.8 Muskingum with Segments

The general equation for this method is:

$$Outflow = C_0 Inflow + C_1 Inflow(-1) + C_2 Outflow(-1)$$

where Outflow and Outflow(-1) are the current and previous outflows respectively as with Inflow and Inflow(-1).

You can further discretize the reach into *n* sub-reaches for better control over attenuation. Note, the routing parameters (either directly as C_0,C_1,and C_2 or as K and X) are specified via the Routing Parameters category and **the same parameters are used for each segment in the reach**.

This routing method is called from the **Solve Outflow** dispatch method. That is, when Muskingum with Segments is selected, the only dispatch method is **Solve Outflow**. This routing method also includes optional Bank Storage, Gain Loss, Outflow Adjust, and Stage calculations. Note, all of these calculations occur after the routing has occurred, based on the routed flow.

SLOTS ASSOCIATED WITH THIS METHOD:

☞ **NUMBER OF SEGMENTS IN REACH**

Type: Scalar
Units: NONE
Description: Number of segments
Information: This parameter determines the number of columns in the Segment Outflow table. In the equations below, the number of segments is represented by the variable *n*.
I/O: Required input
Links: Not linkable

☞ **OUTFLOW BY SEGMENT**

Type: Agg Series
Units: FLOW
Description: Segment outflow. The segments are numbered upstream to downstream. The Outflow to one segment is the inflow to the next segment.
Information: The outflow from the reach is the outflow from the last segment. The columns in this slot will be resized to the number of segments input by the user, and the number of rows will be the number of timesteps in the run. At the beginning of the run, if the initial timestep Outflow by Segment is not

known, the Outflow by Segment is set to the Reach Inflow or Outflow, if known.

Information: Input at Initial timestep, Output at other timesteps
I/O: Output only
Links: Not linkable

METHOD DETAILS:

First, the method checks the following:

- The current timestep's inflow is checked. It should be known or the dispatch method would not have been called.
- If the previous Inflow is not known, the method exits and waits.
- The Outflow by Segment from the previous timestep is checked for validity. If they are all valid from previous solutions, they are used directly and the method continues. If any of the Outflow by Segment are not valid, the following checks are made:

1. If the Outflow is not known and the last Outflow by Segment is known, then Outflow is set to that value. All other Outflow by Segments are also set to that value.
2. If the Outflow is not known, Outflow and each Outflow by Segment is set equal to the Inflow and the routing method exits. This scenario can occur the first time the reach dispatches. This situation is used when you would like the initial Outflow = Inflow.
3. If the Outflow is known (input or set by a rule), each Outflow by Segment is set equal to the Outflow and the method exits. This scenario can occur the first time the reach dispatches and achieves the need where initial Outflow is specified.

Then, the method executes the selected method in the **Routing Parameters** category to determine the C_0,C_1,and C_2 parameters. These C_0,C_1,and C_2 are used by each segment in the reach.

Next, the method sets up arrays of the variables I and O with the known information. The arrays have 2 rows (timesteps) and *n* columns (segments). Of note, I(t, 1) = Inflow. The following table shows the array with the variables. Those next to each other are set to the same value, for example O(t-1,2) = I(t-1,3).

Diagram of discretization for a 3 segment reach:

Timestep	Reach	Segment 1		Segment 2		Segment3		Reach
t-1	Inflow(t-1)	I(t-1,1)	O(t-1,1)	I(t-1,2)	O(t-1,2)	I(t-1,3)	O(t-1,3)	Outflow(t-1)
t	Inflow(t)	I(t,1)	O(t,1)	I(t,2)	O(t,2)	I(t,3)	O(t,3)	Outflow(t)

Next, loop over each segment and compute the outflow for each segment upstream to downstream:

For (j=1 to Number of Segments)

Compute the Outflow from this segment

$$O(t, j) = C_0 I(t, j) + C_1 I(t-1, j) + C_2 O(t-1, j)$$

Set the inflow value for the next segment, I(t,j+1) = O(t,j)

End For

Set the slot **Outflow by Segment[t,j]** to the values stored in the O array.

Execute **Reach Bank Storage** and **Reach GainLoss** method using the routed flow (Outflow in the last segment)

Set the Reach Outflow equal to the last segment's outflow plus Bank Storage Return flow plus Total Gain Loss.

Execute the selected **Outflow Adjustment** method and reset Outflow if necessary

22.1.1.9 Kinematic

This hydraulic routing method is a finite difference solution of the kinematic wave simplification to the St. Venant Equations. It requires a numerical grid to be specified by the user. The numerical approximation tends to result in a negative mass balance error (reaches lose water) in test models. It is recommended that the Kinematic Improved method **HERE (Section 22.1.1.10)** be used to reduce the mass balance error.

Slots Specific to this Method

☛ Delta X Computational Element Length

Type: Scalar
Units: LENGTH
Description: size of the x discretization
Information: This slot must be smaller than the Reach Length
I/O: Required input
Links: Not linkable

☛ Distributed Depth Output

Type: Table Series
Units: LENGTH
Description: depth at upstream end of each element
Information: This value is calculated by the selected Depth to Flow method after calculating the Distributed Flow Output for the current segment at the current timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Celerity Output

Type: Table Series
Units: VELOCITY
Description: celerity(wave speed) at upstream end of each elements
Information: This value is calculated by the selected Depth to Flow method after calculating the Distributed Flow Output for the current segment at the current timestep. Celerity from the previous timestep is used to calculate Distributed Flow Output at the current timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Flow Output

Type: Table Series
Units: FLOW
Description: inflow value at the upstream end of each element
Information: This value is calculated using Celerity from the previous timestep at the current element according to the equation in step 1 below.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed TopWidth Output

Type: Table Series
Units: LENGTH
Description: width of top of water surface
Information: This value is calculated by the selected Depth to Flow method after calculating the Distributed Flow Output for the current segment at the current timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Velocity Output

Type: Table Series
Units: VELOCITY
Description: velocity at upstream end of each element
Information: This value is calculated by the selected Depth to Flow method after calculating the Distributed Flow Output for the current segment at the current timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Volume Output

Type: Table Series
Units: VOLUME
Description: volume contained in each element
Information: The volume in each segment is calculated as

$$segmentVol[t] = segmentVol[t-1] + segmentInflowVol[t] - segmentOutflowVol[t]$$

Because this routing method is a numerical approximation, the segment outflow values will contain numerical error. This numerical error will accumulate in the Distributed Volume Output calculations. Thus this slot can be used to evaluate the overall mass balance error when using the Kinematic method.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Xsectional Area Output

Type: Table Series
Units: AREA
Description: area at upstream end of each element
Information: This value is calculated by the selected Depth to Flow method after calculating the Distributed Flow Output for the current segment at the current timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Numerical Parameters Output

Type: Table
Units: VARIOUS
Description: Columns describing grid and Courant number of each element for the current timestep
Information: The rows of this table store the value of the Courant number with respect to distance. The number of rows used is based on the discretization parameters defined by the user, **Number of each Length/Delta X**. The number of rows in this table is currently limited to 500; therefore, the number of computational elements for each reach must not be greater than 500, i.e. **Delta X** must not be less than **Reach Length**/500. The first two columns are computed at the beginning of run. The Courant number is computed and updated every time the reach dispatches and routes the flow.
I/O: Output only
Links: Not linkable

☛ Reach Length

Type: Scalar
Units: LENGTH
Description: total length of the reach, from upstream to downstream.
Information: This slot must be larger than Delta X
I/O: Required input
Links: Not linkable

☛ Return Flow

Type: Series
Units: FLOW
Description: return flow into the reach
Information: Enters at the bottom of the reach (i.e. is added directly to the outflow calculated by the method).
I/O: Optional; if not input and not linked, it is set to zero.
Links: May be linked to the Return Flow slot on any object, the Outflow slot on a Groundwater Storage object, the Excess GW Outflow slot on the Groundwater Storage object, or the Surface Return Flow slot on a WaterUser.

The solution is a nonlinear numerical approximation to the continuity and momentum equations shown below.

Continuity:

$$\frac{\partial A}{\partial t} + \frac{\partial A}{\partial x} = q$$

Momentum:

$$S_o = S_f$$

For this numerical solution, accuracy increases as the Courant number approaches unity and decreases as the Courant number diverges from unity. The Courant number for each numerical grid section for the current timestep can be observed in the **Numerical Parameters Output** under the column labeled Courant. The Courant number is calculated as:

$$C = c\frac{\Delta t}{\Delta x}$$

where c = wave celerity [L/T], Δt = routing time step[T], and Δx = spatial step size[L].

This hydraulic routing method also requires the user to select from another category of user selectable methods, **Depth to Flow**. The inputs for these methods are described under that user method category.

In the method description below, a superscript refers to the timestep, and a subscript refers to the spatial location (element number) within the reach, upstream to downstream.

Note: This method is restricted to using a non-zero Inflow for the initial timestep. The Kinematic Improved method HERE (Section 22.1.1.10) does not have this restriction.

At the start of the run this method sets all columns of the Distributed Flow Output at the initial timestep (Start Timestep - 1) equal to the initial Reach Inflow (i.e. the flow is the same at every element). It then calls the selected Depth to Flow method to calculate all flow parameters for each element at the initial timestep.

For each timestep within the run, it first sets the flow for the first element:

$$Q_1^t = Inflow - Diversion + LocalInflow$$

If Diversion or Local Inflow are not used, they are set to zero.

Then the selected Depth to Flow method is called calculate all flow parameter values for the first element using Q_1^t. The values are set in the first column of the Distributed Output table series slots.

The method then loops over all remaining elements in the reach, upstream to downstream, to carry out the finite difference approximation in the following steps:

1. Calculate the flow in the given element using the finite difference approximation of the kinematic wave simplification to the St. Venant Equations, which has the form:

$$Q_i^t = \frac{Q_i^{t-1} + Q_{i-1}^t\left(\frac{c_i^{t-1}\Delta t}{\Delta x}\right)}{1 + \frac{c_i^{t-1}\Delta t}{\Delta x}}$$

(This equation is a rearrangement of equation 9.6.6 in Chow et. al., 1988[1] with a small modification to the calculation of celerity.)

2. Call the selected Depth to Flow method to calculate flow parameters using Q_i^t. The values are set in the Distributed Output table series slots. The primary value returned is the celerity (wave speed), c_1^t.

3. Calculate the Distributed Volume Output, the volume of water within the element:

$$V_i^t = V_i^{t-1} + (Q_{i-1}^t - Q_i^t) \times Timestep$$

1. Chow, Ven Te, David R. Maidment, and Larry W. Mays. *Applied Hydrology*. McGraw-Hill, NewYork, 1988.

After calculating flow at the final element, all gains, losses and Return Flow are added to the final element flow to give the total Reach Outflow.

22.1.1.10 Kinematic Improved

This hydraulic routing method is a finite difference solution of the kinematic wave simplification to the St. Venant Equations. It requires a numerical grid to be specified by the user. The Kinematic Improved method is very similar to the Kinematic method. The only difference is in the flow value used to calculate celerity and other distributed output values for each element in the reach as detailed below. This formulation allows for a smaller Delta X Computational Element Length, which in turn reduces overall mass balance error when compared to the Kinematic method.

Slots Specific to this Method

☛ Delta X Computational Element Length

Type: Scalar
Units: LENGTH
Description: size of the x discretization
Information: This slot must be smaller than the Reach Length
I/O: Required input
Links: Not linkable

☛ Distributed Depth Output

Type: Table Series
Units: LENGTH
Description: depth at upstream end of each element
Information: This value is calculated by the selected Depth to Flow method using the average of the previous timestep flow at the current element and the current timestep flow at the upstream element.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Celerity Output

Type: Table Series
Units: VELOCITY
Description: celerity(wave speed) at upstream end of each elements
Information: This value is calculated by the selected Depth to Flow method using the average of the previous timestep flow at the current element and the current timestep flow at the upstream element.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Flow Output

Type: Table Series
Units: FLOW
Description: inflow value at the upstream end of each element
Information:
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed TopWidth Output

Type: Table Series
Units: LENGTH
Description: width of top of water surface
Information: This value is calculated by the selected Depth to Flow method using the average of the previous timestep flow at the current element and the current timestep flow at the upstream element.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Velocity Output

Type: Table Series
Units: VELOCITY
Description: velocity at upstream end of each element
Information: This value is calculated by the selected Depth to Flow method using the average of the previous timestep flow at the current element and the current timestep flow at the upstream element.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Volume Output

Type: Table Series
Units: VOLUME
Description: volume contained in each element
Information: The volume in each segment is calculated as

$$segmentVol[t] = segmentVol[t-1] + segmentInflowVol[t] - segmentOutflowVol[t]$$

Because this routing method is a numerical approximation, the segment outflow values will contain numerical error. This numerical error will accumulate in the Distributed Volume Output calculations. Thus this slot can be used to evaluate the overall mass balance error when using the Kinematic Improved method. The mass balance error can be reduced by reducing the size of Delta X Computational Length.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Xsectional Area Output

Type: Table Series
Units: AREA
Description: area at upstream end of each element
Information: This value is calculated by the selected Depth to Flow method using the average of the previous timestep flow at the current element and the current timestep flow at the upstream element.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Numerical Parameters Output

Type: Table
Units: VARIOUS
Description: Columns describing grid and Courant number of each element for the current timestep
Information: The rows of this table store the value of the Courant number with respect to distance. The number of rows used is based on the discretization parameters defined by the user, **Number of each Length/Delta X**. The number of rows in this table is currently limited to 500; therefore, the number of computational elements for each reach must not be greater than 500, i.e. **Delta X** must not be less than **Reach Length**/500. The first two columns are computed at the beginning of run. The Courant number is computed and updated every time the reach dispatches and routes the flow.
I/O: Output only
Links: Not linkable

☛ Reach Length

Type: Scalar
Units: LENGTH
Description: total length of the reach, from upstream to downstream.
Information: This slot must be larger than Delta X
I/O: Required input
Links: Not linkable

☛ Return Flow

Type: Series
Units: FLOW
Description: return flow into the reach
Information: Enters at the bottom of the reach (i.e. is added directly to the outflow calculated by the method).
I/O: Optional; if not input and not linked, it is set to zero.
Links: May be linked to the Return Flow slot on any object, the Outflow slot on a Groundwater Storage object, the Excess GW Outflow slot on the Groundwater Storage object, or the Surface Return Flow slot on a WaterUser.

The solution is a nonlinear numerical approximation to the continuity and momentum equations shown below.

Continuity:

$$\frac{\partial A}{\partial t} + \frac{\partial A}{\partial x} = q$$

Momentum:

$$S_o = S_f$$

For this numerical solution, the total mass balance error will decrease as the Delta X Computational Element Length (Δx) decreases (number of segments increases). Decreasing Δx will reduce the dispersion in the numerical solution, resulting in higher peaks and lower valleys in the Reach Outflow. Decreasing Δx will also increase the Courant number. The Courant number for each numerical grid section for the current timestep can be observed in the **Numerical Parameters Output** under the column labeled Courant. The Courant number is calculated as:

$$C = c\frac{\Delta t}{\Delta x}$$

where c = wave celerity [L/T], Δt = routing time step[T], and Δx = spatial step size[L].

This hydraulic routing method also requires the user to select from another category of user selectable methods, **Depth to Flow**. The inputs for these methods are described under that user method category.

In the method description below, a superscript refers to the timestep, and a subscript refers to the spatial location (element number) within the reach, upstream to downstream.

At the start of the run this method sets all columns of the Distributed Flow Output at the initial timestep (Start Timestep - 1) equal to the initial Reach Inflow (i.e. the flow is the same at every element). It then calls the selected Depth to Flow method to calculate all flow parameters for each element at the initial timestep.

For each timestep within the run, it first sets the flow for the first element:

$$Q_1^t = Inflow - Diversion + LocalInflow$$

If Diversion or Local Inflow are not used, they are set to zero.

Then the selected Depth to Flow method is called calculate all flow parameter values for the first element using Q_1^t. The values are set in the first columns of the Distributed Output table series slots.

The method then loops over all remaining elements in the reach, upstream to downstream, to carry out the finite difference approximation in the following steps:

1. Calculate the temporary flow value to use for flow parameter calculations:

$$flowTemp = \frac{Q_i^{t-1} + Q_{i-1}^t}{2}$$

2. Call the selected Depth to Flow method to calculate flow parameters using *flowTemp*. The values are set in the Distributed Output table series slots. The primary value returned is the celerity (wave speed), c_1^t.

3. Calculate the flow in the given element using the finite difference approximation. This finite difference scheme is an implicit backward difference solution of the kinematic wave simplification to the St. Venant Equations and has the form of:

$$Q_i^t = \frac{Q_i^{t-1} + Q_{i-1}^t\left(\frac{c_i^t \Delta t}{\Delta x}\right)}{1 + \frac{c_i^t \Delta t}{\Delta x}}$$

(This equation is a rearrangement of equation 9.6.6 in Chow et. al., 1988[1]. Celerity is substituted using equation 9.3.11.)

4. Calculate the Distributed Volume Output, the volume of water within the element:

$$V_i^t = V_i^{t-1} + (Q_{i-1}^t - Q_i^t) \times Timestep$$

After calculating flow at the final element, all gains, losses and Return Flow are added to the final element flow to give the total Reach Outflow.

1. Chow, Ven Te, David R. Maidment, and Larry W. Mays. *Applied Hydrology*. McGraw-Hill, NewYork, 1988.

22.1.1.11 Muskingum Cunge

This hydraulic routing method uses the Muskingum routing equation, but with X and K coefficients that are physically based, as an approximation to the diffusive flow equations.

Background

The standard form of the Muskingum equation, is:

$$O_t = C_0 I_t + C_1 I_{t-1} + C_2 O_{t-1}$$

When applies to a spatially distributed grid, the flow in grid i is:

$$Q_t^i = C_0 Q_t^{i-1} + C_1 Q_{t-1}^{i-1} + C_2 Q_{t-1}^i$$

where t is the incremental timestep at which the calculation is occurring. In the standard Muskingum Cunge equation, C0, C1, and C2 are functions of the Courant number C and Reynolds number D.

$$C_0 = \frac{-1+C+D}{1+C+D} \quad C_1 = \frac{1+C-D}{1+C+D} \quad C_2 = \frac{1-C+D}{1+C+D}$$

C and D are calculated as follows:

$$C = \frac{c(\Delta t)}{\Delta x} \quad D = \frac{Q}{S_o c \Delta x T_w}$$

Q is the grid's flow, S_o is the Energy Slope, c is the wave celerity, and Tw is the top width of the channel calculated for the given flow Q. The wave celerity c and Tw are found using the selected Depth to Flow (HERE (Section 22.1.5)) method for the flow, Q, at the appropriate timestep. The incremental timestep Δt is specified by the user while the spatial distribution Δx is calculated to keep the Courant number close to 1 for the reference discharge.

Following are the slots specific to this method then specific details about the method are described.

Slots Specific to this Method

☛ Delta X Computational Element Length

Type: Scalar
Units: LENGTH
Description: size of the x discretization
Information: Note that this slot is determined by the method, while it is input by the user for the other hydraulic routing methods.
I/O: Output only
Links: Not linkable

☛ DISTRIBUTED CELERITY OUTPUT

Type: Table Series
Units: VELOCITY
Description: celerity(wave speed) at each elements
Information: The value displayed in each row is the value for the last incremental timestep for each run timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ DISTRIBUTED DEPTH OUTPUT

Type: Table Series
Units: LENGTH
Description: depth of each element
Information: The value displayed in each row is the value for the last incremental timestep for each run timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ DISTRIBUTED FLOW OUTPUT

Type: Table Series
Units: FLOW
Description: flow value of each element
Information: The 0th column represents the flow to be routed (Inflow + Local Inflow - Diversion). The last column represents the routed flow and is copied to the outflow. This is actually the flow for the last incremental timestep and is NOT an average over the timestep. Note, in the iteration, the method uses the current (t) and next timestep (t+1) rows to store the previous and current values: current row = previous incremental timestep, next row = current incremental timestep. At the end of each incremental timestep, the value is copied from the next timestep’s row to the current timestep’s row. The only effect visible to the user of this iteration mechanism is that the last row of the table is the same as the last minus one row; the last row is beyond the end of the run and can be ignored.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed MuskingumCunge Output

Type: Table
Units: VARIOUS
Description: holds output parameters of routing method.
Information: C0, C1, C2 are all reported here for the current timestep, along with D, C+D, X and K.
I/O: Output only.
Links: Not linkable

☛ Distributed TopWidth Output

Type: Table Series
Units: LENGTH
Description: width of top of water surface
Information: The value displayed in each row is the value for the last incremental timestep for each run timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Velocity Output

Type: Table Series
Units: VELOCITY
Description: velocity of each element
Information: The value displayed in each row is the value for the last incremental timestep for each run timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Volume Output

Type: Table Series
Units: VOLUME
Description: volume contained in each element
Information: The value displayed in each row is the value for the last incremental timestep for each run timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ DISTRIBUTED XSECTIONAL AREA OUTPUT

Type: Table Series
Units: AREA
Description: area of each element
Information: The value displayed in each row is the value for the last incremental timestep for each run timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ ENERGY SLOPE

Type: Scalar
Units: NO UNITS
Description: slope of the reach from upstream to downstream (vertical/horizontal)
Information: Must be positive, used to calculate the routing distance step
I/O: Required input
Links: Not linkable

☛ EXTREME FLOW VALUES

Type: Table
Units: FLOW
Description: base and max flow values expected during simulation run
Information: Used for calculation of the routing distance step
I/O: Required input
Links: Not linkable

☛ NUMERICAL PARAMETERS OUTPUT

Type: Table
Units: VARIOUS
Description: Columns describing grid and courant number of each element for the current timestep
Information: The rows of this table store the value of the courant number with respect to distance. The number of rows used is based on the discretization parameters, **Number of each Length/Delta X**. The number of rows in this table is currently limited to 500.
I/O: Output only
Links: Not linkable

☛ Reach Length

Type: Scalar
Units: LENGTH
Description: total length of the reach, from upstream to downstream.
Information:
I/O: Required input
Links: Not linkable

☛ Return Flow

Type: Series
Units: FLOW
Description: return flow into the reach
Information: Enters at the bottom of the reach (i.e. is added directly to the outflow calculated by the method).
I/O: Optional; if not input and not linked, it is set to zero.
Links: May be linked to the Return Flow slot on any object, the Outflow slot on a Groundwater Storage object, the Excess GW Outflow slot on the Groundwater Storage object, or the Surface Return Flow slot on a WaterUser.

☛ Routing Time Step

Type: Table
Units: TIME
Description: timestep used by method, Δt in the equations below
Information: This must be smaller than the simulation timestep. The value should be larger than the estimated time of travel for a wave through the reach. It must result in an exact integer number of routing timesteps per simulation timestep.
I/O: Required input
Links: Not linkable

☛ Max Iterations

Type: Scalar Slot
Units: NONE
Description: This values specifies the maximum number of iterations used by convergence algorithm to compute the reach outflow.
I/O: Input; defaults to 20 if not set by the user
Links: Not linkable

Method Details:

The spatial distribution Δx is automatically calculated by RiverWare. The incremental spatial step is determined such that the Courant number, C, will be close to one to reduce the effects of numerical dispersion. Since the discharge will vary for a simulation, the Courant number will also vary. To pick a value for the incremental spatial step that minimizes the effects of numerical dispersion, the user inputs maximum and minimum discharges expected for a simulation in the Extreme Values slot. The average of these two discharges is used as the reference discharge. The wave celerity, c, is then computed with this reference discharge using the selected Depth to Flow (HERE (Section 22.1.5)). The spatial step size, Δx is computed with the input Δt and the Courant number, C set to be 1.0.

$$\Delta x = \frac{c(\Delta t)}{C}$$

The scheme is unconditionally stable for 0.0 <= X <= 0.5 or with D (cell Reynolds number) values less than unity. The cell Reynolds number, D, is defined as the ratio of hydraulic diffusivity to grid diffusivity:

$$D = \frac{Q}{S_o c \Delta x T_w}$$

Where Q is the grid's flow, S_o is the Energy Slope, c is the wave celerity, and T_w is the top width of the channel calculated for the given flow Q. T_w is found using the selected Depth to Flow method. X is related to D via:

$$X = \frac{1}{2}(1 - D)$$

The value of D for each numerical grid section for the current timestep can be observed in the **Distributed MuskingumCunge Output** table under the column labeled D. It should be noted that in this method if the flow result is very small or negative, C and D become 1, and X becomes 0.0. If the above equation for C0 results in a negative value, C and D are set to 1, and C0 is set to 0.

The value of **C + D** for each numerical grid section for the current timestep can be observed in the **Distributed MuskingumCunge Output** table under the column labeled **C + D**.

Method Steps

The method proceeds as follows:

Calculate the total inflow for the timestep (i.e. the inflow plus local inflow minus diversion) to the reach.

The method loops through the incremental timesteps, t, and the spatial grid, i, to find the flow at each element Q_t^i as follows:

FOR EACH incremental timestep, t

The inflow at each incremental timestep is determined by interpolating between the inflow for the previous day and the inflow for the current day. This is then set on the 0th column of the Distributed Flow Output slot

FOR EACH spatial step, i, in the grid

1. Estimate the flow Q_t^i using the three point method. $Q_t^i = \frac{(Q_t^{i-1} + Q_{t-1}^{i-1} + Q_{t-1}^i)}{3}$
2. Using this flow, call the selected Depth to Flow method to determine c_t, Tw_t, A_t.
3. Calculate $D = \frac{Q_t^i}{S_o c \Delta x Tw_t}$ and $C = \frac{c(\Delta t)}{\Delta x}$.

   If the flow is zero, set D and C equal to 1.0.
4. If the following does not hold true: 0.0 <= $X = \frac{1}{2}(1 - D)$ <= 0.5, issue an error and stop the run. The scheme is not stable.
5. Calculate C_0, C_1, and C_2 using:

$$C_0 = \frac{-1 + C + D}{1 + C + D} \quad C_1 = \frac{1 + C - D}{1 + C + D} \quad C_2 = \frac{1 - C + D}{1 + C + D}$$

6. Calculate a new Q_t^i using the Muskingum equation: $Q_t^i = C_o Q_t^{i-1} + C_1 Q_{t-1}^{i-1} + C_2 Q_{t-1}^i$
7. If the new Q_t^i is within convergence of the old Q_t^i, go to the next spatial step, otherwise determine a new Q_t^i using a four point scheme:

$$\frac{(Q_t^{i-1} + Q_{t-1}^{i-1} + Q_{t-1}^i + Q_t^i)}{4}$$

8. Return to step 2 and repeat until the flow, Q_t^i, converges.

END FOR loop over spatial step, i

9. Compute Distributed Volume Output for each element in the reach as the previous incremental timestep's Distributed Volume Output plus the average spatial flow in the reach.

END FOR loop over incremental timestep, t

Using the value in the last column of the Distributed Flow Output slot, execute the bankstorage and gain loss calculations, add in Return Flow, and execute negative outflow adjustment calculations. Each may add or subtract water from the reach. Set the resulting flow on the Outflow slot.

Finally, compute and set the Total Outflow Storage as the sum of the previous Total Outflow Storage plus the Outflow converted to a volume.

Note: To assure the routing parameters are appropriately computed for the given inflow, the reference flow along with corresponding values for the wave celerity and top width are used to compute new values for the cell Reynolds number, D, and the Courant number, C at each timestep. However, changing C and D values in the middle of a simulation results in a volume conservation error. Test results show that this error can be significant for sharply rising and falling hydrograph but the total volume conservation error over a flood event is less than 1%. See the “Muskingum Cunge Improved” HERE (Section 22.1.1.12) for more information on a similar method that better conserves mass.

22.1.1.12 Muskingum Cunge Improved

This hydraulic routing method is an improvement to the “Muskingum Cunge” method in which the coefficients are adjusted to better conserve mass as proposed by Todini, 2007[1].

BACKGROUND

The standard form of the Muskingum equation, is:

$$O_t = C_0 I_t + C_1 I_{t-1} + C_2 O_{t-1}$$

When applies to a spatially distributed grid, the flow in grid i is:

$$Q_t^i = C_0 Q_t^{i-1} + C_1 Q_{t-1}^{i-1} + C_2 Q_{t-1}^i$$

where t is the incremental timestep at which the calculation is occurring. In the standard Muskingum Cunge equation, C0, C1, and C2 are functions of the Courant number C and Reynolds number D.

$$C_0 = \frac{-1+C+D}{1+C+D} \quad C_1 = \frac{1+C-D}{1+C+D} \quad C_2 = \frac{1-C+D}{1+C+D}$$

C and D are calculated as follows:

$$C = \frac{c(\Delta t)}{\Delta x} \quad D = \frac{Q}{S_o c \Delta x T_w}$$

Q is the grid’s flow, S_o is the Energy Slope, c is the wave celerity, and Tw is the top width of the channel calculated for the given flow Q. The wave celerity c and Tw are found using the selected Depth to Flow (HERE (Section 22.1.5)) method for the flow, Q, at the appropriate timestep. The incremental timestep Δt is specified by the user while the spatial distribution Δx is calculated to keep the Courant number close to 1 for the reference discharge.

In the variable parameter, Muskingum Cunge routing method (“Muskingum Cunge”) the cell Reynolds number, D, and the Courant number, C change at each timestep for each cell. Changing C and D values in the middle of a simulation results in a volume conservation error and storage inconsistency at steady state as shown by Todini.

1. Todini, E. 2007, A mass conservation and water storage consistent variable parameter Muskingum-Cunge approach, Hydrology and Earth System Sciences, vol 11, 1645-1659) *http://www.hydrol-earth-syst-sci.net/11/1645/2007/hess-11-1645-2007.pdf*

To mitigate this loss of mass, Todini re-derived the equations (not shown) and calculated the coefficients as:

$$C_0 = \frac{-1 + C^*_t + D^*_t}{1 + C^*_t + D^*_t} \quad C_1 = \frac{1 + C^*_{t-1} - D^*_{t-1}}{1 + C^*_t + D^*_t} \times \frac{C^*_t}{C^*_{t-1}} \quad C_2 = \frac{1 - C^*_{t-1} + D^*_{t-1}}{1 + C^*_t + D^*_t} \times \frac{C^*_t}{C^*_{t-1}}$$

Where C* and D* are the corrected Courant number and Reynolds number, respectively as follows:

$$C^*_{t-1} = \frac{c_{t-1}(\Delta t)}{\beta_{t-1}\Delta x} \quad C^*_t = \frac{c_t(\Delta t)}{\beta_t \Delta x}$$

$$\beta_{t-1} = \frac{c_{t-1}A_{t-1}}{Q_{t-1}} \quad \beta_t = \frac{c_t A_t}{Q_t}$$

$$D^*_{t-1} = \frac{Q_{t-1}}{\beta_{t-1}S_o c_{t-1}\Delta x Tw_{t-1}} \quad D^*_t = \frac{Q_t}{\beta_t S_o c_t \Delta x Tw_t}$$

The “Muskingum Cunge Improved” method is an implementation of this scheme. Although Todini proposed relationships to define the channel parameters for common channel geometry, this method uses an iterative 3 and 4 point scheme and the selected Depth to Flow method to determine these same parameters (area, top width, depth, velocity, and celerity).

In addition, the reach inflow is used as the inflow to each incremental timestep. The Outflow is an average of the outflow over each incremental timestep. Full details of this method follow the slot descriptions.

Slots Specific to this Method

☛ Delta X Computational Element Length

Type: 1X1 table slot
Units: Length
Description: size of the spatial discretization
Information: Note that this slot is determined by the method, while it is input by the user for the other hydraulic routing methods.
I/O: Output only
Links: Not linkable

☛ Distributed Celerity Output

Type: Table Series Slot
Units: Velocity
Description: celerity(wave speed) at each of the elements
Information: The value displayed in each row is the value for the last incremental timestep for each run timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Courant Output

Type: Table Series Slot
Units: None
Description: The adjusted Courant value (C*) for each element in the reach.
Information: The value displayed in each row is the value for the last incremental timestep for each run timestep. Note, in the iteration, the method uses the current and next timestep rows to store the previous and current values: current row = previous incremental timestep, next row = current incremental timestep. At the end of each incremental timestep, the value is copied from the next timestep's row to the current timestep's row. The only effect visible to the user of this iteration mechanism is that the last row of the table is the same as the last minus one row; the last row is beyond the end of the run and can be ignored.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Depth Output

Type: Table Series Slot
Units: Length
Description: depth of each element
Information: The value displayed in each row is the value for the last incremental timestep for each run timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Flow Output

Type: Table Series Slot
Units: Flow
Description: Flow value of each element in the reach.
Information: The 0th column represents the flow to be routed (Inflow + Local Inflow - Diversion). The last column represents the routed flow; but the flow value displayed in each row is the flow in the last incremental timestep for each run timestep. This is not necessarily the same as the average over the run timestep. The Routed Flow slot holds this average over the incremental

timesteps and is then used in subsequent calculations on the reach. Note, in the iteration, the method uses the current and next timestep rows to store the previous and current values: current row = previous incremental timestep, next row = current incremental timestep. At the end of each incremental timestep, the value is copied from the next timestep's row to the current timestep's row. The only effect visible to the user of this iteration mechanism is that the last row of the table is the same as the last minus one row; the last row is beyond the end of the run and can be ignored.

I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed MuskingumCunge Output

Type: Table
Units: Various
Description: holds output parameters of routing method.
Information: C0, C1, C2 are all reported here for the current timestep, along with D, C+D, X and K. Note, C and D are also stored in the Distributed Courant Output and Distributed Reynolds Output slots.
I/O: Output only
Links: Not linkable

☛ Distributed Previous Flow Output

Type: Table Series Slot
Units: Flow
Description: The Distributed Flow Output offset by one timestep
Information: This slot is used to store the previous distributed flow so that if the reach dispatch more than once, the distributed flow is not lost. The value displayed in each row is the value for the last incremental timestep for each run timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not Linkable

☛ Distributed Reynolds Output

Type: Table Series Slot
Units: None
Description: The adjusted Reynolds number (D*) for each element in the reach
Information: The value displayed in each row is the value for the last incremental timestep for each run timestep. Note, in the iteration, the method uses the current and next timestep rows to store the previous and current values: current row = previous incremental timestep, next row = current incremental timestep. At the end of each incremental timestep, the value is copied from the next timestep's row to the current timestep's row. The only effect visible to the user of this iteration mechanism is that the last row of the table is the same as

the last minus one row; the last row is beyond the end of the run and can be ignored.

I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not Linkable

☛ Distributed TopWidth Output
Type: Table Series Slot
Units: LENGTH
Description: width of top of water surface
Information: The value displayed in each row is the value for the last incremental timestep for each run timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Velocity Output
Type: Table Series Slot
Units: VELOCITY
Description: velocity of each element
Information: The value displayed in each row is the value for the last incremental timestep for each run timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Volume Output
Type: Table Series Slot
Units: VOLUME
Description: volume contained in each spatial element of the reach
Information: The value displayed in each row is the value for the last incremental timestep for each run timestep. This volume is the end of timestep value.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Distributed Xsectional Area Output
Type: Table Series Slot
Units: AREA
Description: area of each element
Information: The value displayed in each row is the value for the last incremental timestep for each run timestep.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Energy Slope

Type: Scalar
Units: No Units
Description: slope of the reach from upstream to downstream (vertical/horizontal)
Information: Must be positive
I/O: Required input
Links: Not linkable

☛ Extreme Flow Values

Type: 1X1 Table
Units: Flow
Description: base and max flow values expected during simulation run.
Information: Used for calculation of the routing distance step
I/O: Required input
Links: Not linkable

☛ Max Iterations

Type: 1X1 Table Slot
Units: NONE
Description: This values specifies the maximum number of iterations used by convergence algorithm to compute the reach outflow.
I/O: Input; defaults to 20 if not set by the user
Links: Not linkable

☛ Numerical Parameters Output

Type: Table
Units: Various
Description: Columns describing grid and Courant number of each element for the current timestep
Information: The rows of this table store the value of the courant number with respect to distance. The number of rows used is based on the discretization parameters, Number of each Length/Delta X. The number of rows in this table is currently limited to 500.
I/O: Output only
Links: Not linkable

☛ Previous Outflow

Type: Series Slot
Units: Flow
Description: Previous Outflow
Information: This is the outflow from the reach, offset by one timestep
I/O: Output only
Links: Linkable

☛ Reach Length

Type: 1X1 Table Slot
Units: Length
Description: total length of the reach, from upstream to downstream.
Information:
I/O: Required input
Links: Not linkable

☛ Reach Volume

Type: Series Slot
Units: Volume
Description: The volume of water in the reach at the **end** of the timestep.
Information: This volume is calculated as the sum of the Distributed Volume Output for each row. It is the sum of water over each spatial step along the reach.
I/O: Output only
Links: Not linkable

☛ Return Flow

Type: Series Slot
Units: Flow
Description: return flow into the reach
Information: Enters at the bottom of the reach (i.e. is added directly to the Routed Flow calculated by the method).
I/O: Optional; if not input and not linked, it is set to zero.
Links: May be linked to the Return Flow slot on any object, the Outflow slot on a Groundwater Storage object, the Excess GW Outflow slot on the Groundwater Storage object, or the Surface Return Flow slot on a WaterUser.

☛ Routed Flow

Type: Series Slot
Units: Flow
Description: The flow in the reach after routing has occurred but before gain/loss, return flow or bankstorage is added.
Information: This is calculated by averaging the downstream most distributed outflow from the reach over each of the incremental timesteps
I/O: Output only
Links: Not linkable

☛ ROUTING TIME STEP

Type: 1X1 Table Slot
Units: TIME
Description: timestep used by method, Δt in the equations below
Information: This must be smaller than the simulation timestep. The value should be larger than the estimated time of travel for a wave through the reach. It must result in an exact integer number of routing timesteps per simulation timestep.
I/O: Required input
Links: Not linkable

☛ TOTAL OUTFLOW STORAGE

Type: Series Slot
Units: VOLUME
Description: Cumulative volume of outflow throughout the run.
Information: It is calculated as the previous Total Outflow Storage plus the Outflow converted to a volume
I/O: Output only
Links: Not linkable

METHOD DETAILS

The incremental timestep Δt is related to the spatial distribution Δx as:

$$\Delta x = \frac{c(\Delta t)}{C}$$

Since the discharge varies for a simulation, the Courant number will also vary. To pick a value for the incremental spatial step that minimizes the effects of numerical dispersion, the user inputs minimum and maximum discharges expected for a simulation in the Extreme Flow Values slot. The average of these two discharges is used as the reference discharge when computing the wave celerity, c, using the selected Depth to Flow method. Then, the incremental spatial step Δx is determined with the Courant number, C, set equal to 1.0 to reduce the effects of numerical dispersion. $\Delta x = c(\Delta t)$

Note: This routing method loops over the incremental timestep which must be smaller than the run timestep. The values displayed in the “Distributed Output” slots are the values for the last incremental timestep. The incremental values are strictly internal and there is no way to show the values for intermediate incremental timesteps.

The method proceeds as follows:

Calculate the inflow (i.e. the inflow plus local inflow minus diversion) to the reach and set the flow in the 0th column of the Distributed Flow Output slot equal to this flow. This same inflow value is used for each incremental timestep (no interpolation is used like in the original “Muskingum Cunge” as this leads to a further mass inconsistency). Using this inflow, call the selected Depth to Flow method to determine c, T_w, and A_t. Calculate $D*$ and $C*$ using those values. Set the 0th column of the appropriate table series slot.

The method loops through the incremental timesteps, t, and the spatial grid, i, to find the flow at each element Q_t^i as follows:

FOR EACH incremental timestep, t

FOR EACH spatial step, i, in the grid

1. Estimate the flow Q_t^i using the three point method. $Q_t^i = \frac{(Q_t^{i-1} + Q_{t-1}^{i-1} + Q_{t-1}^i)}{3}$
2. Using this flow, call the selected Depth to Flow method to determine c_t, Tw_t, A_t.
3. Calculate $D*_t = \frac{Q_t^i}{\beta_t S_o c_t \Delta x Tw_t}$ and $C*_t = \frac{c_t(\Delta t)}{\beta_t \Delta x}$ where $\beta_t = \frac{c_t A_t}{Q_t^i}$.

   If the flow is zero, set $D*_t$ and $C*_t$ equal to 1.0.
4. If the following does not hold true: 0.0 <= $X = \frac{1}{2}(1 - D*_t)$ <= 0.5, issue an error and stop the run. The scheme is not stable.
5. Calculate C_0, C_1, and C_2 using:

$$C_0 = \frac{-1 + C*_t + D*_t}{1 + C*_t + D*_t} \quad C_1 = \frac{1 + C*_{t-1} - D*_{t-1}}{1 + C*_t + D*_t} \times \frac{C*_t}{C*_{t-1}} \quad C_2 = \frac{1 - C*_{t-1} + D*_{t-1}}{1 + C*_t + D*_t} \times \frac{C*_t}{C*_{t-1}}$$

   Note that $D*_{t-1}$ and $C*_{t-1}$ are from the previous incremental timestep and were stored in the appropriate cell of the respective table series slot.
6. Calculate a new Q_t^i using the Muskingum equation: $Q_t^i = C_o Q_t^{i-1} + C_1 Q_{t-1}^{i-1} + C_2 Q_{t-1}^i$
7. If the new Q_t^i is within convergence of the old Q_t^i, go to the next spatial step, otherwise determine a new Q_t^i using a four point scheme:

$$\frac{(Q_t^{i-1} + Q_{t-1}^{i-1} + Q_{t-1}^i + Q_t^i)}{4}$$

8. Return to step 2 and repeat until the flow, Q_t^i, converges. Once convergence occurs, set a temporary variable tempOutflowVolume to track the total volume of outflow (last column of the Distributed Flow Output slot, i.e. the bottom of the reach). Add to this total volume at each incremental timestep.

END FOR loop over spatial step, i

9. Compute Distributed Volume Output for each element in the reach:

$$Q_t^{i-1}\frac{(1-D^*{}_t)\Delta t}{2C^*{}_t} + Q_t^i\frac{(1+D^*{}_t)\Delta t}{2C^*{}_t}$$

END FOR loop over incremental timestep, t

Next, the Routed Flow is calculated as the average outflow over the incremental timesteps. This is calculated by dividing the tempOutflowVolume by the run timestep. Then, sum up the Distributed Volume Output over space (i.e. a given row) and set this value on the Reach Volume slot.

Using the Routed Flow, execute the bankstorage and gain loss calculations, add in Return Flow, and execute negative outflow adjustment calculations. Each may add or subtract water from the reach. Set the resulting flow on the Outflow slot.

Finally, compute and set the Total Outflow Storage as the sum of the previous Total Outflow Storage plus the Outflow converted to a volume.

This method was implemented to achieve better mass conservation but is not guaranteed to fully conserve mass. Because the user selects the Extreme Flow values and Routing Timestep, there are circumstances where flow is not fully conserved. This is especially true when there are abrupt flow changes and flows that are zero. Testing has shown that mass is fully conserved in well behaved problems but not necessarily in real-world problems. Regardless, the mass conservation is much better than the original routing method.

22.1.1.13 MacCormack

This hydraulic routing method is a finite difference solution of the kinematic wave simplification to the St. Venant Equations. It requires a numerical grid to be specified by the user.

Slots Specific to this Method

☛ Delta X Computational Element Length

Type: Scalar
Units: LENGTH
Description: size of the x discretization
Information: This slot must be smaller than the Reach Length.
I/O: Required input
Links: Not linkable

☛ DISTRIBUTED CELERITY OUTPUT

Type: Table Series
Units: VELOCITY
Description: celerity(wave speed) at upstream end of each elements
Information: This is a temporary slot that is not saved in the model file.
I/O: Output only
Links: Not linkable

☛ DISTRIBUTED DEPTH OUTPUT

Type: Table Series
Units: LENGTH
Description: depth at upstream end of each element
Information: This is a temporary slot that is not saved in the model file.
I/O: Output only
Links: Not linkable

☛ DISTRIBUTED FLOW OUTPUT

Type: Table Series
Units: FLOW
Description: inflow value at the upstream end of each element
Information: This is a temporary slot that is not saved in the model file.
I/O: Output only
Links: Not linkable

☛ DISTRIBUTED TOPWIDTH OUTPUT

Type: Table Series
Units: LENGTH
Description: width of top of water surface
Information: This is a temporary slot that is not saved in the model file.
I/O: Output only
Links: Not linkable

☛ DISTRIBUTED VELOCITY OUTPUT

Type: Table Series
Units: VELOCITY
Description: velocity at upstream end of each element
Information: This is a temporary slot that is not saved in the model file.
I/O: Output only
Links: Not linkable

☛ Distributed Volume Output

Type: Table Series
Units: VOLUME
Description: volume contained in each element
Information: This is a temporary slot that is not saved in the model file.
I/O: Output only
Links: Not linkable

☛ Distributed Xsectional Area Output

Type: Table Series
Units: AREA
Description: area at upstream end of each element
Information: This is a temporary slot that is not saved in the model file.
I/O: Output only
Links: Not linkable

☛ Numerical Parameters Output

Type: Table
Units: VARIOUS
Description: Columns describing grid and courant number of each element for the current timestep
Information: The rows of this table store the value of the courant number with respect to distance. The number of rows used is based on the discretization parameters defined by the user, **Number of each Length/Delta X**. The number of rows in this table is currently limited to 500; therefore, the number of computational elements for each reach must not be greater than 500, i.e. **Delta X** must not be less than **Reach Length**/500.
I/O: Output only
Links: Not linkable

☛ Reach Length

Type: Scalar
Units: LENGTH
Description: total length of the reach, from upstream to downstream.
Information: This slot must be larger than Delta X
I/O: Required input
Links: Not linkable

☛ Return Flow

Type: Series
Units: FLOW
Description: return flow into the reach
Information: Enters at the bottom of the reach (i.e. is added directly to the outflow calculated by the method).
I/O: Optional; if not input and not linked, it is set to zero.
Links: May be linked to the Return Flow slot on any object, the Outflow slot on a Groundwater Storage object, the Excess GW Outflow slot on the Groundwater Storage object, or the Surface Return Flow slot on a WaterUser.

☛ Routing Time Step

Type: Table
Units: TIME
Description: timestep used by method
Information: This must be smaller than the simulation timestep. The value should be larger than the estimated time of travel for a wave through the reach.
I/O: Required input
Links: Not linkable

The solution is a nonlinear numerical approximation to the continuity and momentum equations shown below.

Continuity:

$$\frac{\partial A}{\partial t} + \frac{\partial A}{\partial x} = q$$

Momentum:

$$S_o = S_f$$

For this numerical solution, accuracy increases as the Courant number approaches unity and decreases as the Courant number diverges from unity. The Courant number for each numerical grid section for the current timestep can be observed in the **Numerical Parameters Output** under the column labeled Courant. The Courant number is calculated as:

$$C = c\frac{\Delta t}{\Delta x}$$

where c = wave celerity [L/T], Δt = routing time step[T], and Δx = spatial step size[L].

This hydraulic routing method also requires the user to select from another category of user selectable methods: **Depth to Flow**. The inputs for these methods are described under that user method category.

The algorithm uses is a predictor corrector scheme that is unstable for a Courant number greater than unity. The advantage of this scheme is that it is second order accurate and can minimize numerical diffusion. The general equations for this method are:

predictor:

$$Q_i^{\overline{t+1}} = (Q_i^t + Q_{t+1}^t + Q_t^t)\frac{c_i^t \Delta t}{\Delta x}$$

corrector:

$$Q_i^{\overline{t+1}} = Q_i^t + 0.5(Q_{i+1}^t + Q_i^t)\frac{c_i^t \Delta t}{\Delta x} + (Q_i^{\overline{t+1}} + Q_{i-1}^{\overline{t+1}}) + \frac{\left(c_i^{\overline{t+1}}{}_t\right)\Delta t}{\Delta x}$$

22.1.1.14 Storage Routing

This method is a simple storage method that solves for outflows given current and previous inflow values. The reach is broken up into a user-specified number of linked segments and flows are calculated for each segment.

Note: This method is based on an empirical formula that uses a numeric approximation; therefore the method does not guarantee that mass balance will be preserved exactly.

Slots Specific to this Method

☛ Number of Segments in Reach

Type: Scalar
Units: NOUNITS
Description: number of segments upstream to downstream
Information: This will determine the number of columns in the Segment Outflow table as well.
I/O: Required input
Links: Not linkable

☛ Segment Outflow

Type: Table Series
Units: FLOW
Description: segment outflow
Information: The outflow from the reach is the outflow from the last segment. The columns in this table will be resized to the number of segments input by the user, and the number of rows will be the number of timesteps in the run.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Storage Time Coefficient

Type: Scalar
Units: NOUNITS
Description: value that is divided by the result of the average flow and exponent to arrive at time in storage
Information: The units of this slot should be $Volume^{exponent}$ (a value should be used that is in $(ft^3)^{exponent}$). This coefficient may be determined by trial and error, and should not be negative. The value must correspond to a flow value in cfs and storage time in hours, regardless of the user units on other slots.
I/O: Required input
Links: Not linkable

☛ Storage Time Exponent

Type: Scalar
Units: NOUNITS
Description: exponent of mean flow value.
Information: Usually between -1 and 1. The value must correspond to a flow value in cfs and storage time in hours, regardless of the user units on other slots.
I/O: Required input
Links: Not linkable

The algorithm proceeds in this fashion:

- If the previous Inflow value is not known, the method exits.
- The outflow value for each segment from the previous timestep is checked for validity. There are then three possible scenarios:

   1. If the segment outflows are not valid and previous Outflow is not valid, Outflow is set equal to inflow plus gain loss and the method exits.
   2. If the segment outflows are not valid, and previous Outflow is valid, set all segment outflows equal to
      previous Inflow, and continue the routing method.

3. If the segment outflows are valid, continue the method.

- Find the mean interior flow from the previous timestep as the average of all segment outflows.
- Find the time of storage in the reach based on the following empirical formula(in cfs):

$$storagetime = \frac{coefficient}{(meanfl)^{exponent}}$$

where storagetime is the time of storage *in hours*, coefficient & exponent are user input constants in the slots above, and meanfl is the average interior flow of the previous timestep. (The storagetime calculation is always made with meanfl in units of cfs regardless of the user units on any flow slots. The values for Storage Time Coefficient and Storage Time Exponent should be set accordingly.) The time in storage can be used as a “conversion” from storage to outflow:

$$Storage = storagetime \bullet Outflow$$

- Find the number of routing phases, n. If the time of storage is greater than half of the simulation timestep, n is 1. Otherwise, n is calculated as:

$$n = \frac{(timestep)}{(2storagetime)} + 1$$

- If n is greater than 48 from this equation, n is set to 6.
- The inflow into the first segment for each routing phase is

$$segInflow = \frac{inflow + inflow(-1)}{2n}$$

- For each routing phase, the outflow from each segment is:

$$segOutflow = \frac{(segInflow - segOutflow(-1))timestep}{\frac{timestep}{2} + storagetime} + segOutflow(-1)$$

- This routing equation is based on the storage-outflow relation, storagetime, from above, and the continuity equation:.

$$\left(\frac{Inflow + Inflow(-1)}{2} - \frac{Outflow + Outflow(-1)}{2}\right)timestep = Storage - Storage(-1)$$

- The inflow into the next segment is the average of the segments previous and current outflows

The current timestep's Outflow for the object is equal to the last segment's outflow.

22.1.1.15 Variable Storage Routing

This method is a simple storage method that solves for outflows given current and previous inflow values. The reach is broken up into a user-specified number of linked segments and flows are calculated for each segment. This method differs from the Storage Routing method only in the determination of the storagetime value. In this method, the storage time exponent and coefficient are found from a table lookup based on flow instead of constant values.

Note: This method is based on an empirical formula that uses a numeric approximation; therefore the method does not guarantee that mass balance will be preserved exactly.

Slots Specific to this Method

☛ Flow Range

Type: Table
Units: FLOW
Description: the ranges of flow rates corresponding to the Variable Storage Time Tables
Information: The number of flow ranges allowed in this table is currently limited to 10. The columns of this table represent, in ascending order, the maximum flow rate for a specific flow range starting with 0 as the base. These columns are labeled **Flow Range 1 - 10**. Therefore, the first column represents a range of flows from 0 to the value input by the user in the **Flow Range 1** column. The last flow range entered by the user should be a flow rate greater than any anticipated for the simulation. Otherwise, a value of 0.0 will be used for the **Variable Storage Time** slots for flows outside of this range. It is not necessary to utilize all 10 columns of this table. Use only the columns needed to designate the desired flow ranges.
I/O: Required input
Links: Not linkable

☛ Number of Segments in Reach

Type: Scalar
Units: NOUNITS
Description: number of segments upstream to downstream
Information: This will determine the number of columns in the Segment Outflow table as well.
I/O: Required input
Links: Not linkable

☛ Segment Outflow

Type: Table Series
Units: FLOW
Description: segment outflow
Information: The outflow from the reach is the outflow from the last segment. The columns in this table will be resized to the number of segments input by the user, and the number of rows will be the number of timesteps in the run.
I/O: Output only - This is a temporary slot that is not saved in the model file.
Links: Not linkable

☛ Variable Storage Time Coefficient

Type: Series
Units: NOUNITS
Description: value that is divided by the result of the average flow and exponent to arrive at time in storage
Information: The units of this slot should be $Volume^{exponent}$ (a value should be used that is in $(ft^3)^{exponent}$).
I/O: Output only
Links: Not linkable

☛ Variable Storage Time Coefficient Table

Type: Table
Units: NOUNITS
Description: Variable Storage Time Coefficient for each specific Flow Range
Information: The columns of the table correspond to the flow ranges defined in **Flow Range**. The values must correspond to a flow value in cfs and storage time in hours, regardless of the user units on other slots.
I/O: Required input
Links: Not linkable

☛ Variable Storage Time Exponent

Type: Series
Units: NOUNITS
Description: exponent of mean flow value.
Information: Usually between -1 and 1.
I/O: Output only
Links: Not linkable

☛ Variable Storage Time Exponent Table

Type: Table
Units: NOUNITS
Description: Variable Storage Time Exponent for each specific Flow Range
Information: The columns of the table correspond to the flow ranges defined in **Flow Range.** The values must correspond to a flow value in cfs and storage time in hours, regardless of the user units on other slots.
I/O: Required input
Links: Not linkable

The algorithm proceeds in this fashion:

- If the previous Inflow value is not known, the method exits.
- The outflow value for each segment from the previous timestep is checked for validity. There are then three possible scenarios:

  1. If the segment outflows are not valid and previous Outflow is not valid, Outflow is set equal to inflow
  plus gain loss and the method exits.

  2. If the segment outflows are not valid, and previous Outflow is valid, set all segment outflows equal to previous Inflow, and continue the routing method.

  3. If the segment outflows are valid, continue the method.

- Find the mean interior flow from the previous timestep as the average of all segment outflows.
- Find the time of storage in the reach based on the following empirical formula(in cfs):

$$storagetime = \frac{coefficient}{(meanfl)^{exponent}}$$

  where storagetime is the time of storage *in hours*, coefficient & exponent are user input constants in the slots above, and meanfl is the average interior flow of the previous timestep. (The storagetime calculation is always made with meanfl in units of cfs regardless of the user units on any flow slots. The values in the Variable Storage Time Coefficient Table and Variable Storage Time Exponent Table should be set accordingly.) The time in storage can be used as a “conversion” from storage to outflow:

$$Storage = storagetime \bullet Outflow$$

- Find the number of routing phases, n. If the time of storage is greater than half of the simulation timestep, n is 1. Otherwise, n is calculated as:

$$n = \frac{timestep}{2storagetime} + 1$$

- If n is greater than 48 from this equation, n is set to 6.
- The inflow into the first segment for each routing phase is

$$segInflow = \frac{inflow + inflow(-1)}{2n}$$

For each routing phase, the outflow from each segment is:

$$segOutflow = \frac{(segInflow - segOutflow(-1))timestep}{\frac{timestep}{2} + storagetime} + segOutflow(-1)$$

- This routing equation is based on the storage-outflow relation, storagetime, from above, and the continuity equation:

$$\left(\frac{Inflow + Inflow(-1)}{2} - \frac{Outflow + Outflow(-1)}{2}\right)timestep = Storage - Storage(-1)$$

- The inflow into the next segment is the average of the segments previous and current outflows

The current timestep's Outflow for the object is equal to the last segment's outflow.

22.1.1.16 Modified Puls

The **Modified Puls** algorithm solves for the Reach **Outflow** given current and previous Inflow and Storage values using the following continuity equation:

$$\left(\frac{Inflow + Inflow(t-1)}{2} - \frac{Outflow + Outflow(t-1)}{2}\right)\Delta t = Storage - Storage(t-1)$$

Rewritten, this equation is:

$$\frac{Storage}{\Delta t} + \frac{Outflow}{2} = \frac{Inflow + Inflow(t-1)}{2} + \left(\frac{Storage(t-1)}{\Delta t} - \frac{Outflow(t-1)}{2}\right)$$

The equation has two unknowns, Storage and Outflow, both on the left side. In the Modified Puls method, the user specifies a tabular relationship of Storage versus Outflow in the **Storage Outflow Table**. Given this table of data, the set of Storage and Outflow can be found to satisfy the equation.

You can further discretize the reach into *n* sub-reaches for better control over the attenuation of the flood wave. The Storage Outflow Table is specified for the entire reach. Therefore, during the solution, the Storage in each subreach is assumed to be 1/n of the total storage. Each subreach is assumed to be identical and uniform.

This routing method is called from the **Solve Storage Routing Outflow** dispatch method. That is, when Modified Puls is selected, the only available dispatch method is **Solve**

Storage Routing Outflow. This dispatch method solves downstream and includes optional Bank Storage, Gain Loss, Outflow Adjust, Stage, and Volume calculations. Note, all of these calculations occur after the routing has occurred, based on the routed flow.

SLOTS ASSOCIATED WITH THIS METHOD:

☛ NUMBER OF SEGMENTS IN REACH

Type: Scalar
Units: NONE
Description: Number of segments
Information: This parameter determines the number of columns in the Segment Outflow table. The travel time through a segment should be approximately equal to the timestep size. An estimate is to divide the reach length by the product of the wave celerity and timestep size. In the equations below, the number of segments is represented by the variable *n*.
I/O: Required input
Links: Not linkable

☛ OUTFLOW BY SEGMENT

Type: Agg Series
Units: FLOW
Description: Segment outflow. The segments are numbered upstream to downstream. The Outflow to one segment is the inflow to the next segment.
Information: The outflow from the reach is the outflow from the last segment. The columns in this slot will be resized to the number of segments input by the user, and the number of rows will be the number of timesteps in the run. At the beginning of the run, if the initial timestep Outflow by Segment is not known, the Outflow by Segment is set to the Reach Inflow or Outflow, if known.
Information: Input at Initial timestep, Output at other timesteps
I/O: Output only
Links: Not linkable

☛ STORAGE BY SEGMENT

Type: Agg Series
Units: VOLUME
Description: Segment storage. The segments are numbered upstream to downstream, left to right.
Information: The columns in this table will be resized to the number of segments input by the user, and the number of rows will be the number of timesteps in the run.
I/O: Output only
Links: Not linkable

☛ Storage Outflow Table

Type: Table
Units: Volume vs Flow
Description: The table relating Storage in the entire reach to the Outflow.
Information: The data must be monotonically increasing and cover the entire range of storage and flow values expected.
I/O: Input only
Links: Not Linkable

☛ Storage Outflow Indication Table

Type: Table
Units: Volume vs Flow vs Flow
Description: This generated table repeats the Storage Outflow Table and then shows a third column for the Indication variable.
Information: This table is generated at the beginning of run by looking up each Storage and Outflow on the Storage Outflow Table. These become the first and second columns. The third column, the Indication, is computed by looking up each row's Storage and Outflow and then computing the value:

$$\frac{Storage}{\Delta t \times n} + \frac{Outflow}{2}$$

Information: Thus, the Indication is for a single segment. All segments are assumed to be the same. In addition, the timestep length is assumed to be constant.
I/O: Output only. This table is read only.
Links: Not Linkable

Method Details:

First, the method checks the following:

- The current timestep's inflow is checked. It should be known or the dispatch method would not have been called.
- If the previous Inflow is not known, the method exits and waits.
- The Outflow by Segment from the previous timestep is checked for validity. If they are all valid from previous solutions, they are used directly and the method continues. If any of the Outflow by Segment are not valid, the following checks are made:

1. If the Outflow is not known and the last Outflow by Segment is known, then Outflow is set to that value. All other Outflow by Segments are also set to that value.
2. Else, if the Outflow is not known, Outflow and each Outflow by Segment is set equal to the Inflow and the routing method exits. This scenario can occur the first time the reach dispatches. This situation is used when you would like the initial Outflow = Inflow.

3. If the Outflow is known (input or set by a rule), each Outflow by Segment is set equal to the Outflow and the method exits. This scenario can occur the first time the reach dispatches and achieves the need where initial Outflow is specified.

If the previous Storage values for each segment are not known, the method computes the previous Storage from the **Storage Outflow Table**.

Next, the method sets up arrays of the variables I, S, and O with the known information. The arrays will have 2 rows (timesteps) and *n* columns (segments). Of note, I(t, 1) = Inflow. The following table shows the array with the variables. Those next to each other are set to the same value, for example O(t-1,2) = I(t-1,3).

Diagram of discretization for a 3 segment reach:

Timestep	Reach	Segment 1			Segment 2			Segment3			Reach
t-1	Inflow(t-1)	I(t-1,1)	S(t-1,1)	O(t-1,1)	I(t-1,2)	S(t-1,2)	O(t-1,2)	I(t-1,3)	S(t-1,3)	O(t-1,3)	Outflow(t-1)
t	Inflow(t)	I(t,1)	S(t,1)	O(t,1)	I(t,2)	S(t,2)	O(t,2)	I(t,3)	S(t,3)	O(t,3)	Outflow(t)

Next, loop over each segment and compute the outflow and storage for each segment upstream to downstream:

For (j=1 to Number of Segments)

Compute the StorOutIndication variable

$$StorOutIndication = \frac{I(t,j) + I(t-1,j)}{2} + \left(\frac{S(t-1,j)}{\Delta t \times n} - \frac{O(t-1,j)}{2}\right)$$

If StorOutIndication is less than zero, reset it to zero.

Look up the StorOutIndication on the **Storage Outflow Indication Table** to find the Storage and Outflow.

Record the values found as S(t,j) and O(t,j).

Set the inflow value for the next segment, I(t,j+1) = O(t,j)

End For

Set the slots **Storage by Segment[t,j]**, **Outflow by Segment[t,j]** to the values stored in the arrays.

Execute **Reach Bank Storage** and **Reach GainLoss** method using the routed flow (Outflow in the last segment)

Set the Reach Outflow equal to the last segment's outflow plus Bank Storage Return flow plus Total Gain Loss.

Execute the selected **Outflow Adjustment** method and reset Outflow if necessary

Return to the **Solve Storage Routing Outflow** dispatch method.

22.1.2 Routing Parameters

This method category is dependent upon the selection of Muskingum outing in the Routing category. It enables the user to specify either the Muskingum K and X values or the C0, C1 and C2 coefficients. There is also the option of providing this data as a times series.

22.1.2.1 None

This is the default method. It performs no calculations. An error is posted in the beginning of the run if this method is selected.

22.1.2.2 Input K and X Values

This method is used to specify constant Muskingum K and X values.

Slots Specific to this Method

MuskingumCoefficientK

Type: Scalar Slot
Units: TIME
Description: lag time of the centroid of a flood wave
Information:
I/O: Required input
Links: Not linkable

MuskingumCoefficientX

Type: Scalar Slot
Units: NONE
Description: attenuation factor
Information:
I/O: Required input
Links: Not linkable

The Muskingum Coefficient K and Muskingum Coefficient X are used to calculated the C0, C1 and C2 coefficients by the following equations:

$$C_o = \frac{\Delta t - 2KX}{\Delta t + 2K(1 - X)}$$

$$C_1 = \frac{\Delta t + 2KX}{\Delta t + 2K(1 - X)}$$

$$C_2 = \frac{-\Delta t + 2K(1 - X)}{\Delta t + 2K(1 - X)}$$

where Δt is the simulation timestep, K[Time] and X[no units] are the Muskingum coefficients.

22.1.2.3 Input Coefficients

This method is used to input the characteristic Muskingum coefficients directly. The coefficients are not allowed to vary with time.

Slots Specific to this Method

☛ C0 C1 C2 Coefficients

Type: Table
Units: NOUNITS
Description: a table containing the three characteristic Muskingum coefficients
Information: A value for each coefficient is given in the first three columns of the table.
I/O: Required input
Links: Not linkable

The values in the C0 C1 C2 Coefficients slot are used directly in the Muskingum routing calculations.

22.1.2.4 Input Time Series K and X

This method is used to specify time variable data for K and X.

Slots Specific to this Method

☛ Muskingum K Time Series

Type: Series
Units: TIME
Description: lag time of the centroid of a flood wave
Information:
I/O: Required input
Links: Not linkable

☛ Muskingum X Time Series

Type: Series
Units: NONE
Description: attenuation factor
Information:
I/O: Required input
Links: Not linkable

The K value and X value are used to calculated the C0, C1 and C2 coefficients by the following equations:

$$C_o = \frac{\Delta t - 2KX}{\Delta t + 2K(1-X)}$$

$$C_1 = \frac{\Delta t + 2KX}{\Delta t + 2K(1-X)}$$

$$C_2 = \frac{-\Delta t + 2K(1-X)}{\Delta t + 2K(1-X)}$$

where Δt is the simulation timestep, K[Time] and X[no units] are the Muskingum coefficients.

22.1.2.5 Input Time Series Coefficients

This method is used to specify the characteristic Muskingum coefficients as a time series.

Slots Specific to this Method

☛ C0 Time Series

Type: Series
Units: NOUNITS
Description: a time series range of C0 values
Information:
I/O: Required input
Links: Not linkable

☛ C1 Time Series

Type: Series
Units: NOUNITS
Description: a time series range of C1 values
Information:
I/O: Required input
Links: Not linkable

☛ C2 Time Series

Type: Series
Units: NOUNITS
Description: a time series range of C2 values
Information:
I/O: Required input
Links: Not linkable

The C0, C1 and C2 values are used directly in the Muskingum routing calculations.

22.1.3 Local Inflow and Solution Direction

The Local Inflow and Solution Direction methods determine how the Local Inflow is calculated. This method category is available for all routing methods except Storage Routing and Variable Storage Routing. This category can also affect how the simulation dispatches (the solution direction). The user can limit what dispatch methods are available.

22.1.3.1 No Local Inflow, Solve Inflow or Outflow

This method is the default for the category, and should be selected when modeling of local inflow is not desired and the model should be able to solve for either inflow or outflow. There are no slots specifically associated with this method.

22.1.3.2 Specify Local Inflow, Solve Inflow or Outflow

This method treats Local Inflow as an optional input. It is not available for the Storage Routing or Variable Storage Routing methods.

When this method is selected, the Generate Local Inflows category becomes visible. Within this category, methods are available to generate the Local Inflow.

Slots Specific to this Method

☛ **Local Inflow**

Type: Series
Units: FLOW
Description: local inflow into the reach
Information: Local Inflow is added to Inflow then the resulting flow is routed. Local Inflow is not included in the calculation of Available for Diversion.
I/O: Optional; if this slot is not linked and not input, it is set to zero.
Links: May be linked to the Outflow slot on any object, or an Expression Slot or Series Slot on a Data Object.

☛ **Local Inflow Adjust**

Type: Series
Units: FLOW
Description: adjustment made to the local inflow
Information: Only used with the No Routing method. The functionality of this slot is discussed in the No Routing dispatch methods.
I/O: Output only
Links: Not linkable

22.1.3.3 Solve Inflow, Outflow or Local Inflow

This method is only available for the No Routing method, and allows local inflow to be a third variable in the Routing method. This method changes the dispatch conditions for the No Routing method so that two of the following parameters; Inflow, Outflow and Local Inflow must be known to solve for the third.

Slots Specific to this Method

☛ **Local Inflow**

Type: Series
Units: FLOW
Description: local inflow into the reach
Information:
I/O: Optional; if this slot is not input, it will be solved for when both Inflow and Outflow are known. If it is input, when either Inflow or Outflow are known, the other will be solved for.
Links: May be linked to the Outflow slot on any object, or an Expression Slot or Series Slot on a Data Object.

☛ **Local Inflow Adjust**

Type: Series
Units: FLOW
Description: adjustment made to the local inflow
Information: Only used with the No Routing method when Outflow or Local Inflow are solved for. The functionality of this slot is discussed in the No Routing dispatch methods.
I/O: Optional
Links: Not linkable

22.1.3.4 Contingent Local Inflow or Solve Outflow

This method is only available for the No Routing method, and allows local inflow to be a third variable in the Routing method when inflow and outflow are both known. This method changes the dispatch conditions for the No Routing method, so that either Inflow and Outflow must be known to solve for Local Inflow, or Inflow must be known to solve for the Outflow (and Local Inflow is assumed to be zero). This method is designed to allow for the use of non-continuous gaged data for reconciliation when it is available. It may be used by supplying Inflow for all timesteps, and supplying gage data at an object below the reach when it is available as outflow, and leaving the unknown data as NaN.

Slots Specific to this Method

☛ **LOCAL INFLOW**

Type: Series
Units: FLOW
Description: local inflow into the reach
Information:
I/O: Optional; if this slot is not input but inflow and outflow are known, it will be solved for. If outflow is not known, the reach will solve for outflow given inflow assuming local inflow is zero. Or, the slot can be input and the reach will solve for outflow given inflow and local inflow.
Links: May be linked to the Outflow slot on any object, or a Series Slot on a Data Object.

☛ **LOCAL INFLOW ADJUST**

Type: Series
Units: FLOW
Description: adjustment made to the local inflow
Information: This slot is only used with the No Routing method.
I/O: Optional
Links: Not linkable

22.1.3.5 Solve Local Inflow or Outflow

This method is only available for the No Routing method, and allows local inflow to be a third variable in the Routing method when inflow and outflow are both known. This method changes the dispatch conditions for the No Routing method, so that either Inflow and Outflow must be known to solve for Local Inflow, or Inflow and Local Inflow must be known to solve for the Outflow.

SLOTS SPECIFIC TO THIS METHOD

☛ **LOCAL INFLOW**

Type: Series
Units: FLOW
Description: local inflow into the reach
Information:
I/O: Optional; if this slot is not specified but inflow and outflow are known, it will be solved for. Or, the slot can be specified and the reach will solve for outflow given inflow and local inflow.
Links: May be linked to the Outflow slot on any object, or a Series Slot on a Data Object.

☛ Local Inflow Adjust

Type: Series
Units: FLOW
Description: adjustment made to the local inflow
Information: This slot is only used with the No Routing method.
I/O: Optional
Links: Not linkable

22.1.3.6 Specify Local Inflow, Solve Outflow

This method is available for the No Routing and Time Lag routing methods, and allows Local Inflow to be a second variable in the Routing method when Inflow is known. This method changes the available dispatch conditions. For the No Routing method, the available dispatch method is solveNROutflow. For Time Lag routing, the available dispatch method is solveTLOutflowDSOnly.

When this method is selected, the Generate Local Inflows category becomes visible. Within this category, methods are available to generate the Local Inflow.

Slots Specific to this Method

☛ Local Inflow

Type: Series
Units: FLOW
Description: local inflow into the reach
Information:
I/O: Optional; if this slot is not input, it will default to zero. If it is input the reach will solve for outflow given inflow and Local Inflow.
Links: May be linked to the Outflow slot on any object, or an Expression Slot or Series Slot on a Data Object.

☛ Local Inflow Adjust

Type: Series
Units: FLOW
Description: adjustment made to the local inflow
Information: This slot is only used with the No Routing method.
I/O: Optional
Links: Not linkable

22.1.3.7 No Local Inflow, Solve Outflow

This method is available for the No Routing and Time Lag routing methods. This method changes the available dispatch conditions. For the No Routing method, the available dispatch method is solveNROutflow. For Time Lag routing, the available dispatch method is solveTLOutflowDSOnly.

SLOTS SPECIFIC TO THIS METHOD: NONE

22.1.3.8 Local Inflow MODFLOW Return

Note: RiverWare's connection with MODFLOW is currently not functional. This method has been disabled and cannot be selected. An error will be posted at model load if this method was previously selected. Contact CADSWES for help.

This method is available when the "Link to MODFLOW" method is selected from the "MODFLOW Link Category Reach", HERE (Section 22.1.20). The Local Inflow MODFLOW Return is added to the flow equations in the No Routing, Time Lag, Variable Time Lag, Muskingum Cunge, and Muskingum Cunge Improved methods. This method should be used in conjunction with the computational subbasin MODFLOW link methods, HERE (Section 7.2.1).

SLOTS SPECIFIC TO THIS METHOD

☛ LOCAL INFLOW MODFLOW RETURN

Type: Series Slot
Units: FLOW
Description: Surface water return flow entering the Reach. This inflow is from a surface water body represented in MODFLOW.
Information: Local Inflow comes in at the top of the Reach, and is added to the inflow before it is routed.
I/O: Output only
Links: Linkable

A return flow from a surface water body (e.g. riverside drain) represented in MODFLOW to a RiverWare Reach may be desired. When the "Local Inflow MODFLOW Return" method and the "Link to MODFLOW" method are selected, a surface water return flow may be transferred from MODFLOW to the RiverWare Local Inflow MODFLOW Return slot through the computational subbasin framework. The MODFLOW segment source for the return flow may be specified on the computational subbasin, HERE (Section 7.1.28).

22.1.4 Generate Local Inflows

This category contains methods that forecast the Local Inflow based on known inflow values. The user inputs the historical inflows to the reach and the methods adjust those values to represent a forecast. The methods in this category execute at the beginning of each timestep. This category is dependent on having either the Specify Local Inflow, Solve Inflow or Outflow; or the Specify Local Inflow, Solve Outflow method selected.

22.1.4.1 None

This method is the default for this category but does nothing.

22.1.4.2 Geometric Recession

On each timestep in the forecast period, this method will adjust the inflow hydrographs, input in the Deterministic Local Inflow slot, and write the results to the Local Inflow series slot.

Slots Specific to this Method

☛ **Deterministic Local Inflow**

Type: Series
Units: FLOW
Description: This slots holds a timeseries of the actual Local Inflows to the reach. These values are then adjusted by the forecast method and set on the Local Inflow slot.
Information: At the end of the run, the Local Inflow slot will be identical to this slot.
I/O: Input only
Links: Not linkable

☛ **Period of Perfect Knowledge**

Type: Scalar
Units: FLOW
Description: Number of timesteps for which the forecast will equal the Deterministic Local Inflow, i.e., the forecast is known with complete certainty.
Information: Minimum value of 1; maximum value equal to the number of timesteps in the forecast period.
I/O: Input only
Type: Not linkable

☛ **Recession Factor**

Type: Scalar
Units: NONE
Description: A decimal value that is multiplied by the previous Local Inflow to determine the current value after the Period of Perfect Knowledge.
Information:
I/O: Input only
Links: Not linkable

The Deterministic Local Inflow slot values are input for each timestep on a reach. At the beginning of each timestep, the Geometric Recession method is selected. For each forecast timestep within the period of perfect knowledge, the Local Inflow is set to the Deterministic Local Inflow value. For each forecast timestep after the period of perfect knowledge, the Local Inflow is set by multiplying the value of the Local Inflow from the previous timestep by the constant recession factor.

A value for the Deterministic Local Inflow slot must be known at every timestep during the run. If forecasted Local Inflow values are desired past the end of the run, there must also be values in the Deterministic Local Inflow slot at timesteps past the end of the run. If values for Deterministic Local Inflow are not entered past the end of the run, the Local Inflows for these timesteps are assumed to be zero.

22.1.4.3 Exponential Recession

On each timestep in the forecast period, this method will adjust the inflow hydrographs, input in the Deterministic Local Inflow slot, and write the results to the Local Inflow series slot. The slot values of the Local Inflow will be the same as those in the Deterministic Local Inflow slot during the period of perfect knowledge. After the period of perfect knowledge, the Local Inflow slot values are recessed by a recession equation using the Deterministic Local Inflow slot value on the last day of perfect knowledge.

SLOTS SPECIFIC TO THIS METHOD

☛ Deterministic Local Inflow

Type: Series
Units: FLOW
Description: This slots holds a timeseries of the actual Local Inflows to the reach. These values are then adjusted by the forecast method and set on the Local Inflow slot.
Information: At the end of the run, the Local Inflow slot will be identical to this slot.
I/O: Input only
Links: Not linkable

☛ Forecast Period

Type: Table
Units: NONE
Description: Number of timesteps, not including the current timestep, that the inflow hydrograph will be adjusted.
Information:
I/O: Input only
Links: Not linkable

☛ Period of Perfect Knowledge

Type: Scalar
Units: FLOW
Description: Number of timesteps for which the forecast will equal the Deterministic Local Inflow, i.e., the forecast is known with complete certainty.
Information: Minimum value of 1; maximum value equal to the number of timesteps in the forecast period.
I/O: Input only
Links: Not linkable

☛ Minimum Forecasted Flow

Type: Series
Units: FLOW
Description: The minimum forecasted flow.
Information: If the computed value for Local Inflow is less than the Minimum Forecasted Flow, it is set to the Minimum Forecasted Flow.
I/O: Input only
Links: Not linkable

☛ Low Flow Threshold

Type: Scalar
Units: FLOW
Description: The flow rate that dictates whether to use the Low Flow Recession Coefficient or the High Flow Recession Coefficient.
Information:
I/O: Input only
Links: Not linkable

☛ Low Flow Recession Coefficient

Type: Scalar
Units: NONE
Description: The recession coefficient used when the Deterministic Local Inflow (at the end of the Period of Perfect Knowledge) is below or equal to the Low Flow Threshold.
Information:
I/O: Input only
Links: Not linkable

☛ High Flow Recession Coefficient

Type: Scalar
Units: NONE
Description: The recession coefficient used when the Deterministic Local Inflow (at the end of the Period of Perfect Knowledge) is above the Low Flow Threshold.
Information:
I/O: Input only
Links: Not linkable

The Deterministic Local Inflow slot values are input for each timestep on a reach. At the beginning of each timestep, the Exponential Recession method is selected. For each forecast timestep within the period of perfect knowledge, the Local Inflow is set to the Deterministic Local Inflow value. For each forecast timestep after the period of perfect knowledge, the Local Inflow slot is set as described below:

$$ForecastedFlow = MAX\left[MinimumForecastedFlow, \left(DeterministicLocalInflow \bullet e^{\frac{(-C)t}{T}}\right)\right]$$

where Deterministic Local Inflow is the value in the Deterministic Local Inflow slot at the end of the period of perfect knowledge, C is the recession coefficient, t is the elapsed time of the forecast period, and T is the total time from the end of the period of perfect knowledge to the end of the forecast period.

If Deterministic Local Inflow at the end of the period of perfect knowledge is negative, the Local Inflow at that timestep is exactly equal to the Deterministic Local Inflow. However, the Deterministic Local Inflow used in the recession equation is the last positive value for Deterministic Local Inflow. In the event that there is not a positive value for Deterministic Local Inflow, RiverWare issues a warning, and all values for Local Inflow within the forecast period will be set to the Minimum Forecasted Flow.

A value for the Deterministic Local Inflow slot must be known at every timestep during the run. If forecasted Local Inflow values are desired past the end of the run, there must also be values in the Deterministic Local Inflow slot at timesteps past the end of the run. If values for Deterministic Local Inflow are not entered past the end of the run, the Local Inflows for these timesteps are assumed to be zero.

22.1.4.4 Coefficient and Exponent

On each timestep in the forecast period, this method will adjust the inflow hydrographs, input in the Deterministic Local Inflow slot, and write the results to the Local Inflow series slot.

Slots Specific to this Method

☛ Forecast Inflow Parameters

Type: Table
Units: None
Description: Table slot that contains four parameters used in the forecast inflow method. The first row contains the values for the increasing hydrograph, the second row contains values for the decreasing hydrograph. The first column contains coefficients, the second column contains exponents.
Information: 2X2 table
I/O: Input only
Links: not linkable

☛ Deterministic Local Inflow

Type: Series
Units: FLOW
Description: This slots holds a timeseries of the actual Local Inflows to the reach. These values are then adjusted by the forecast method and set on the Local Inflow slot.
Information: At the end of the run, the Local Inflow slot will be identical to this slot. The logic below uses the **Lower Bound** on the Deterministic Local Inflow slot as

a minimum value. This is specified slot configuration (View->Configure menu). Consider setting this value as needed.

I/O: Input Only
Links: not linkable

FORECAST PERIOD

Type: Table
Units: NONE
Description: Number of timesteps, not including the current timestep, that the inflow hydrograph will be adjusted.
Information:
I/O: Input only
Links: not linkable

The method works as follows: on the current timestep, the Local Inflow is set equal to the Deterministic Local Inflow. The method then loops through the remaining timesteps in the forecast period and sets the Local Inflow using the following formula starting at i = 1:

$$LI_i = LI_{i-1} + LI_{i-1}\frac{(KI_i - KI_{i-1})}{KI_{i-1}}((C^i)^E)$$

where KI_i is the Deterministic Local Inflow at timestep i, LI_i is the Local Inflow at timestep i, C is a coefficient and E is an exponent. The counter i represents the timestep beyond the current timestep. So, i = 1 is the current timestep + 1, i = 2 is the current timestep + 2, etc. The coefficient, C, and exponent, E, are the values in the Forecast Inflow Parameters slot.

NOTE: In the above formulas, there is a mathematical problem if KI_{i-1} is zero or very small. In this situation, the Local Inflow is set to the known inflow at the appropriate index. This allows the simulation to continue with reasonable values for the local inflow. The logic uses the Lower Bound on the Deterministic Local Inflow slot as the minimum value. This is specified on the slot configuration (View->Configure menu). If this value is specified it is checked, otherwise only 0.0 is used in the check. The logic is:

If (Deterministic Local Inflow(i-1) = 0.0 OR

ABS(Deterministic Local Inflow(i-1)) < ABS(Lower Bound)))

Local Inflow(i) = Deterministic Local Inflow(i)

A value for the Deterministic Local Inflow slot must be known at every timestep during the run. If forecasted Local Inflow values are desired past the end of the run, there must also be values in the Deterministic Local Inflow slot at timesteps past the end of the run. If values for Deterministic Local Inflow are not entered past the end of the run, the Local Inflows for these timesteps are assumed to be zero.

Because this method is executed on each timestep and sets Local Inflow values at future timesteps, the Local Inflow slot at pre-simulation timesteps never get set. For lagged reaches, this can be a problem. To solve this, the user must input the correct values in the Local Inflow slot for the necessary pre-simulation timesteps.

22.1.5 Depth to Flow

This category is only available for the hydraulic routing methods: Kinematic, Muskingum Cunge, Muskingum Cunge Improved, and MacCormack routing. The methods in this category control how flow is translated into velocity, depth, area, top width and celerity for use in the Routing methods.

22.1.5.1 None

This method is the default for this category. It will result in an error if it is selected and a run is started.

22.1.5.2 Manning Trapezoid

This method models the channel geometry as a trapezoid.

Slots Specific to this Method

☛ **Bottom Width**

Type: Scalar
Units: LENGTH
Description: width of the bottom of the channel
Information:
I/O: Required input
Links: Not linkable

☛ **Energy Slope**

Type: Scalar
Units: NO UNITS
Description: Slope of the reach from upstream to downstream (vertical/horizontal)
Information: Must be positive
I/O: Required input
Links: Not linkable

☛ **Mannings Roughness n**

Type: Scalar
Units: NO UNITS
Description: Manning's roughness n, empirical representation of the roughness of the channel
Information:
I/O: Required input
Links: Not linkable

☛ **SIDE SLOPE**

Type: Scalar
Units: NO UNITS
Description: Slope of the sides of the channel, S:1 (horizontal:vertical)
Information:
I/O: Required input
Links: Not linkable

This method uses Newtons method to find a depth corresponding to a given flow. See any open channel flow book for the equations relating flow and depth to the rest of the parameters for a trapezoidal channel. This method does not produce any output directly, rather the Routing method sets some output slots based on the results of this method.

22.1.5.3 Manning Wide Rectangle

This method models the channel as a wide rectangle.

☛ **SLOTS SPECIFIC TO THIS METHOD**

☛ **BOTTOM WIDTH**

Type: Scalar
Units: LENGTH
Description: width of the bottom of the channel
Information:
I/O: Required input
Links: Not linkable

☛ **ENERGY SLOPE**

Type: Scalar
Units: NO UNITS
Description: Slope of the reach from upstream to downstream (vertical/horizontal)
Information: Must be positive
I/O: Required input
Links: Not linkable

☛ **MANNINGS ROUGHNESS N**

Type: Scalar
Units: NO UNITS
Description: Manning’s roughness n, empirical representation of the roughness of the channel
Information:
I/O: Required input
Links: Not linkable

This is the simplest geometry to model. This method assumes that channel depth is negligible relative to channel width, and thus top width and the wetted perimeter are assumed to be equal to the bottom width. This method does not produce any output directly, rather the Routing method sets some output slots based on the flow parameter values calculated by this method.

If flow is zero, then all flow parameter values are zero. Otherwise the following values are calculated:

First the flow parameters α and β are set (see, for example, Chow et. al., 1988[1], p. 283),

$$\beta = 0.6$$

$$\alpha = \left(\frac{nP^{2/3}}{\sqrt{S_0}}\right)^{\beta}$$

where n is the Manning's roughness coefficient, P is the wetted perimeter (here equal to the bottom width), and S_0 is the energy slope.

$$Area = \alpha Q^{\beta}$$

where Q is the flow value passed in from the Routing method calling this method.

$$Celerity = \frac{1}{\alpha\beta Q^{\beta - 1}}$$

$$Velocity = \frac{|Q|}{Area}$$

$$Depth = \frac{Area}{BottomWidth}$$

$$TopWidth = BottomWidth$$

22.1.5.4 Power Function

This method models the channel geometry as separate functions relating flow to area, area to depth, and flow to top width. Velocity and celerity are derived from the values of these functions.

Note: The exponential calculations are carried out in ***RiverWare internal units***. Flow is in cms, lengths are in m, areas are in m^2, and velocity is in m/s. The values set in the Beta parameter slots, as set by the user, must be consistent with these units.

1. Chow, Ven Te, David R. Maidment, and Larry W. Mays. *Applied Hydrology*. McGraw-Hill, NewYork, 1988.

Slots Specific to this Method

☛ Alpha for Area to Flow
Type: Scalar
Units: NO UNITS
Description: coefficient in area/flow conversion
Information:
I/O: Required input
Links: Not linkable

☛ Alpha for Depth to Area
Type: Scalar
Units: NO UNITS
Description: coefficient in depth/area conversion
Information:
I/O: Required input
Links: Not linkable

☛ Alpha for Top Width to Flow
Type: Scalar
Units: NO UNITS
Description: coefficient in topwidth/flow conversion
Information:
I/O: Required input
Links: Not linkable

☛ Beta for Area to Flow
Type: Scalar
Units: NO UNITS
Description: exponent in area/flow conversion
Information: The value in this slot must correspond to flow in cms and area in m^2.
I/O: Required input
Links: Not linkable

☛ Beta for Depth to Area
Type: Scalar
Units: NO UNITS
Description: exponent in depth/area conversion
Information: The value in this slot must correspond to flow in area in m^2 and depth in m.
I/O: Required input
Links: Not linkable

☛ Beta for Top Width to Flow

Type: Scalar
Units: NO UNITS
Description: exponent in topwidth/flow conversion
Information: The value in this slot must correspond to flow in cms and TopWidth in m.
I/O: Required input
Links: Not linkable

The equations involved in this method are:

$$Area = AlphaArea(flow^{BetaArea})$$

$$Depth = AlphaDepth(area^{BetaDepth})$$

$$TopWidth = AlphaTopWidth(flow^{BetaTopWidth})$$

All of the calculations are carried out in ***RiverWare internal units*** (SI units), and parameter values (with no units) should be set accordingly.

22.1.6 Stage

This category controls the calculation of river “stage” or elevation. Depending on the method selected, the upstream, downstream and/or average stage may be calculated. Please note that if stage values are linked or used together with other elevation values (as in surface water / ground water interfacing applications), RiverWare assumes that they all share the same datum.

22.1.6.1 None

This method is the default for this category. It performs no calculations. There are no slots specifically associated with this method.

22.1.6.2 Inflow Stage Table Look Up

This method finds the stage in the river from a table interpolation using the inflow. The user may select this method if Head Based Diversion is selected in the Diversion From Reach category. This method is only available when the Reach solves in the downstream direction only. As a result, for “No Routing” and “Time Lag”, this method is available only if the “Specify Local Inflow, Solve Outflow” or “No Local Inflow, Solve Outflow” method is selected. For the other Routing methods, the method is always available as these only solve downstream.

Slots Specific to this Method

☛ Inflow Stage

Type: Series
Units: LENGTH
Description: Stage of reach
Information:
I/O: Output only
Links: May be linked to the Tailwater Base Elevation of a reservoir. In this case, the Reach computes the Inflow Stage which then propagates to the linked slot.

☛ Inflow Stage Table

Type: Table
Units: FLOW versus LENGTH
Description: table to find reach stage from inflow
Information: In reality, this should represent the average stage in the reach
I/O: Required input
Links: Not linkable

☛ **Stage**

Type: Series
Units: Length
Description: the average reach water surface elevation
Information: Values in this slot are output only, except in the case of a time lag in the Reach/Drain Elev Interp method of the Drain Water Surface Elevation method category. This requires that values for Stage are input for pre-dispatching timesteps needed by the time lag.
I/O: Output only
Links: Not linkable

Inflow Stage is calculated by using the Inflow with the Inflow Stage Table. The Stage slot, which is used in certain seepage calculations, is set equal to the Inflow Stage.

22.1.6.3 Stage Table Look Up

This method finds the stage in the river from a table interpolation. The user must select this method if Head Based Diversion is selected in the Diversion From Reach category.

Slots Specific to this Method

☛ **Inflow Stage**

Type: Series
Units: LENGTH
Description: Stage of reach at upstream end
Information:
I/O: Output only
Links: May be linked to the Tailwater Base Elevation of a reservoir. In this case, the Reach computes the Inflow Stage which then propagates to the linked slot.

☛ **Inflow Stage Table**

Type: Table
Units: FLOW versus LENGTH
Description: table to find reach stage from flow at upstream end of reach
Information:
I/O: Required input
Links: Not linkable

☛ OUTFLOW STAGE

Type: Series
Units: LENGTH
Description: Stage of reach at downstream end
Information:
I/O: Output only
Links: May be linked to the Pool Elevation of a reservoir. In this case, the linked slot should compute the value which will then propagate to the Outflow Stage, for use in the Stage computation.

☛ OUTFLOW STAGE TABLE

Type: Table
Units: FLOW VERSUS LENGTH
Description: table to find reach stage from flow at downstream end of reach
Information:
I/O: Required input
Links: Not linkable

☛ STAGE

Type: Series
Units: LENGTH
Description: the average reach water surface elevation
Information: Values in this slot are output only, except in the case of a time lag in the Reach/Drain Elev Interp method of the Drain Water Surface Elevation method category. This requires that values for Stage are input for pre-dispatching timesteps needed by the time lag.
I/O: Output only
Links: Not linkable

Inflow Stage and Outflow Stage are calculated by using the Inflow with the Inflow Stage Table and the Outflow with the Outflow Stage Table, respectively. The Stage slot, which is used in certain seepage calculations, is set equal to an average value whenever possible. If inflow and outflow stage values are known for the current time step, the stage is set as the average of these two values. If one of the values is not known for the current time step, a value is sought at the previous time step. The stage is then set as the average of a current time step stage value (either inflow or outflow) and the previous time step stage value (either inflow or outflow, corresponding to whichever is not available at the current time step). If a value from the previous time step is sought for but is not available--at the first time step of a model run, for example--no averaging is done. The stage slot value is set equal to whichever value is known at the current time step.

IMPORTANT: When the reach is dispatched with solveNRInflow, solveNROutflow, or solveOutflow, the Stage Table Look Up user method is solved twice: once near the very beginning of dispatch and once near the very end. When the object dispatches the first time (for a given time step):

- first the stage is calculated averaging a current flow value and a previous flow value,
- then other user methods are solved, including seepage methods that may depend on the stage value,
- then the flow at the other end of the reach is determined (inflow for solveNRInflow or outflow for solveNROutflow and solveOutflow),
- and finally the stage is recalculated averaging available values for the current time step for both inflow and outflow.

Under this arrangement, the intermediate user methods will use an appropriately averaged value of stage for solution, but the final value in the stage slot is unlikely to represent the value used by those user methods. Also, any subsequent re-dispatch of the objects will cause Stage Table Look Up to average existing current inflow and outflow values at the beginning of dispatch. That value still may not equal the value found in the stage slot at the end of dispatch. Note the stage method "Inflow Stage Table Look Up" HERE (Section 22.1.6.2) does not have this same dispatching pattern and should be considered if you wish to avoid this.

Double solution of the Stage Table Look Up user method as described above does not occur in solveNRLocalInflow, solveTLInflow, solveTLOutflow, solveTLInflowOutflow, solveTLOutflowDSOnly, or solveStorTimeOutflow dispatch methods.

22.1.6.4 Input Stage

The Input Stage Method allows the user to input the average water surface elevation of the Reach for use in certain Seepage calculations.

Slots Specific to the Method

☛ Stage

Type: Series
Units: Length
Description: The average reach water surface elevation
Information:
I/O: Required input
Links: Not linkable

22.1.6.5 Gage Based Stage

The Gage Based Stage method is used when a reach must be discretized, as may be the case with Gradient Seepage Methods. When gages are not available on the Reach element more distant gages must be used to determine the stage.

Slots Specific to this Method

☛ Stage
Type: Series
Units: LENGTH
Description: The average reach water surface elevation
Information:
I/O: Output only
Links: Not linkable

☛ Upstream Gage Elev
Type: Series
Units: LENGTH
Description: The elevation of the upstream gauge
Information: This is the elevation of a gauge which is not immediately adjacent to the Reach. It is some distance upstream.
I/O: Required input
Links: Not linkable

☛ Downstream Gage Elev
Type: Series
Units: LENGTH
Description: The elevation of the downstream gauge
Information: This is the elevation of a gauge which is not immediately adjacent to the Reach. It is some distance downstream.
I/O: Required input
Links: Not linkable

☛ Distance Between Gages
Type: Table
Units: LENGTH
Description: The horizontal distance between upstream and downstream gauges
Information:
I/O: Required input
Links: Not linkable

☛ Downstream Distance to Reach
Type: Table
Units: LENGTH
Description: The horizontal distance from the upstream gauge to the point on the Reach for which elevation is desired
Information:
I/O: Required input
Links: Not linkable

The difference between the upstream and downstream gage readings (the rise) divided by the distance between the gages (the run) provides the Reach gradient. When this is multiplied by the distance downstream from the upstream gage to the Reach element, it provides the change in elevation. This is subtracted from the upstream elevation for the Avg Stage.

22.1.7 Stage Adjustment

This category is only available when the Inflow Stage Table Look Up or Stage Table Lookup method is selected from the Stage category.

22.1.7.1 None

This method is the default for this category. It performs no calculations. There are no slots specifically associated with this method.

22.1.7.2 Stage Coefficient Weighting

This method adjusts the current stage to be a weighted sum of the previously computed current stage and previous timesteps' stage values.

SLOTS SPECIFIC TO THIS METHOD

☛ STAGE WEIGHTING COEFFICIENTS

Type: Table
Units: NONE
Description: Coefficients used to weight previous temporary stage.
Information: The values in the table should sum to 1.0.
I/O: Required Input
Links: N/A

Stage Weighting Coefficients are defined in this table slot which contains a number of user-defined coefficients corresponding to the number of timesteps the user wants to include in the weighted sum. Previously computed Stage values are looked up for the current timestep and a number of previous timesteps, corresponding to the number of coefficients in the aforementioned table. Then, the adjusted Stage is computed as:

$$\text{Stage} = C_0 \times tempStage(t) + C_1 \times Stage(t-1) + \ldots + C_n \times Stage(t-n)$$

Where the C_0, C_1,..., C_n are the **Stage Weighting Coefficients**.

If the **Inflow Stage Table Look Up** method is selected from **Stage** category, the tempStage(t) is the **InflowStage(t).**

If the **Stage Table Look Up** method is selected from **Stage** category, the tempStage(t) is the computed stage, which is typically the average of the **InflowStage(t)** and **OutflowStage(t).**

This method requires previous timestep values. At the start of the run (start to start+n), if these values are not available, the method will use only current timestep tempStage and assume coeffs are {1,...}. At timesteps after (start+n), if the previous values are invalid, a message will be posted and stop the run.

The default convergence on stage may be too loose for this method to work, depending on the magnitudes of stage values. The convergence percentage may have to be tightened from the default 0.01% to an order of magnitude less, 0.001%, or to an even more precise convergence percentage. Convergence is adjusted in the **Configure** menu as described **HERE (Slots.pdf, Section 4.1.4)**.

22.1.8 Volume

This category is only available for the Time Lag and Storage Routing methods.

22.1.8.1 None

This method is the default for this category. It performs no calculations. There are no slots specifically associated with this method.

22.1.8.2 Volume Table Lookup

This method finds the volume in the reach with a mass balance equation. An initial value for Reach Volume must be input by the user.

SLOTS SPECIFIC TO THIS METHOD

☛ **REACH VOLUME**

Type: Series
Units: VOLUME
Description: holds the volume of water in the reach at the current timestep

Note: Must be given a value at the initial timestep of the run in order to execute.

I/O: Output only
Links: Not linkable

The mass balance equation is:

$$ReachVolume = ReachVolume_{-1} + InflowVOL - OutflowVOL$$

22.1.9 Apply Gain Loss

This method category controls how the (Variable) GainLoss Coeff is applied to find the Outflow. This category is only available for the Muskingum and Muskingum Cunge routing methods.

22.1.9.1 Current Outflow

This method applies the (Variable) GainLoss Coeff to the total routed outflow, which is the way that the GainLoss Coeff is applied to most Routing methods. There are no slots specifically associated with this method.

The equation for the final outflow is:

$$Outflow = (RoutedFlow)(1 + (Variable)GainLossCoeff) + (Variable)GainLoss$$

22.1.9.2 Previous Flow Average

This method applies the (Variable) GainLoss Coeff to the average of the previous timestep's inflow and outflow. This is done in order to allow the Muskingum cunge and the Muskingum method to use the same equations for loss.

There are no slots specifically associated with this method.

The equation for the final outflow is:

$$Outflow = (RoutedFlow) + ((Variable)GainLossCoeff)\left(\frac{Inflow_{-1}}{2} + \frac{Outflow_{-1}}{2}\right)$$

$$+ (Variable)GainLoss$$

22.1.10 Gain Loss

This method category is used to model gains and/or losses in the Reach. It is available, as well as all the user methods contained within it, for all Routing methods. Note that the equations given for each Gain Loss method may be slightly modified if the Previous Flow Average method is selected in the Apply Gain Loss category.

22.1.10.1 None

This is the default method. No calculations are performed and there are no slots associated with this method. It should be used if the user does not wish to model gains or losses.

22.1.10.2 Constant Gain Loss

Used when the gain or loss parameters do not vary with respect to time.

Slots Specific to this Method

☛ GainLoss

Type: Table
Units: FLOW
Description: represents a given amount of flow that is gained or lost regardless of the flow rate in the reach
Information: A positive number represents a gain while a negative number represents a loss.
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

☛ GainLoss Coeff

Type: Table
Units: DECIMAL
Description: represents a given percentage of flow in the reach which is gained or lost
Information: Must be a number between -1 and 1. For example, a value of -0.1 means that 10% of the flow in the reach is lost. If solving upstream, must be greater than -1.
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

☛ **TOTAL GAINLOSS**

Type: Series
Units: FLOW
Description: represents the total amount of flow that is gained or lost in a reach for a given timestep
Information: Applied to the bottom of the reach (it is added to the outflow value calculated from the Routing method).
I/O: Output only
Links: Not linkable

$$TotalGainLoss = GainLossCoeff \bullet Routedflow + GainLoss$$

22.1.10.3 Variable Gain Loss

Used when a time series range of gain or loss parameters is required.

SLOTS SPECIFIC TO THIS METHOD

☛ **VARIABLE GAINLOSS**

Type: Series
Units: FLOW
Description: represents a given amount of flow that is gained or lost regardless of the flow rate in the reach
Information: A positive number represents a gain while a negative number represents a loss.
I/O: Optional; defeats to 0.0 if not input.
Links: Not linkable

☛ **VARIABLE GAINLOSS COEFF**

Type: Series
Units: DECIMAL
Description: represents a given percentage of flow in the reach which is gained or lost
Information: Must be a number between -1 and 1. For example, a value of -0.1 means that 10% of the flow in the reach is lost. If solving upstream, must be greaterthan -1.
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

☛ **TOTAL GAINLOSS**

Type: Series
Units: FLOW
Description: represents the total amount of flow that is gained or lost in a reach for a given timestep
Information: Applied to the bottom of the reach (it is added to the outflow value calculated from the Routing method).
I/O: Output only
Links: Not linkable

$$TotalGainLoss = VariableGainLossCoeff \bullet Routedflow + VariableGainLoss$$

22.1.10.4 Seasonal Gain Loss Flow Table

Used to calculate the gain/loss parameters based on season and a range of flow rates.

SLOTS SPECIFIC TO THIS METHOD

☛ **DATE RANGE**

Type: Table
Units: NO UNITS
Description: the actual days of the 366 day year that correspond to the Number of Seasons
Information: Each column of this table corresponds to a particular season and are labeled Season 1 - 12. The two rows labeled 0 and 1 respectively represent the beginning day and ending day of the season column. The allowable range for these inputs is 1 - 366. It is recommended to cover the entire year within the Number of Seasons columns quantified in that slot.
I/O: Required input
Links: Not linkable

☛ **FLOW RANGE**

Type: Table
Units: FLOW
Description: the ranges of flow rates corresponding to the Variable GainLoss Coeff Table and Variable GainLoss Table
Information: The number of flow ranges allowed in this table is currently limited to 10. The columns of this table represent, in ascending order, the maximum flow rate for a specific flow range starting with 0 as the base. These columns are labeled **Flow Range 1 - 10**. Therefore, the first column represents a range of flows from 0 to the value input by the user in the **Flow Range 1** column. The last flow range entered by the user should be a flow rate greater than any anticipated for the simulation. Otherwise, a value of 0.0 will be used for

flows outside of this range. It is not necessary to utilize all 10 columns of this table. Use only the columns needed to designate the desired flow ranges.

I/O: Required input
Links: Not linkable

☛ Number of Season

Type: Scalar
Units: NO UNITS
Description: the number of seasons corresponding to the Variable GainLoss Coeff Table and Variable GainLoss Table
Information: The maximum number of seasons currently allowed in **RiverWare™** is 12.
I/O: Required input
Links: Not linkable

☛ Variable GainLoss

Type: Series
Units: FLOW
Description: represents a given amount of flow that is gained or lost regardless of the flow rate in the reach
Information: A positive number represents a gain while a negative number represents a loss.
I/O: Optional; if not input, it is obtained from the Variable GainLoss Table.
Links: Not linkable

☛ Variable GainLoss Coeff

Type: Series
Units: DECIMAL
Description: represents a given percentage of flow in the reach which is gained or lost
Information: Must be a number between -1 and 1. For example, a value of -0.1 means that 10% of the flow in the reach is lost. If solving upstream, must be greaterthan -1.
I/O: Optional; if not input, it is obtained from the Variable GainLoss Coeff Table.
Links: Not linkable

☛ Variable GainLoss Coeff Table

Type: Table
Units: TIME
Description: Variable GainLoss Coeff values for each specific Date Range and Flow Range
Information: The columns of the table correspond to the flow ranges defined in **Flow Range,** and the rows correspond to the seasons defined in **Date Range**.
I/O: Required input
Links: Not linkable

☛ **VARIABLE GAINLOSS TABLE**

Type: Table
Units: TIME
Description: Variable GainLoss values for each specific Date Range and Flow Range
Information: The columns of the table correspond to the flow ranges defined in **Flow Range,** and the rows correspond to the seasons defined in **Date Range**.
I/O: Required input
Links: Not linkable

☛ **TOTAL GAINLOSS**

Type: Series
Units: FLOW
Description: represents the total amount of flow that is gained or lost in a reach for a given timestep
Information: Applied to the bottom of the reach (it is added to the outflow value calculated from the Routing method).
I/O: Output only
Links: Not linkable

$$TotalGainLoss = VariableGainLossCoeff \bullet Routedflow + VariableGainLoss$$

22.1.10.5 Interpolated Flow Gain Loss

This method is similar to the Seasonal Gain Loss Flow Table method except the data is based on the day of the year and the tables are interpolated to find values for gainloss and the gainloss coefficient.

SLOTS SPECIFIC TO THIS METHOD

☛ **INTERPOLATED GAINLOSS**

Type: Series
Units: FLOW
Description: represents a given amount of flow that is gained or lost regardless of the flow rate in the reach
Information: A positive number represents a gain while a negative number represents a loss. It is calculated by double interpolation of the Interpolated GainLoss Table
I/O: Output only
Links: Not linkable

☛ Interpolated GainLoss Coeff

Type: Series
Units: FRACTION
Description: represents a given percentage of flow in the reach which is gained or lost
Information: Must be a number between -1 and 1. For example, a value of -0.1 means that 10% of the flow in the reach is lost. It is calculated by double interpolation of the Interpolated GainLoss Coeff Table.
I/O: Output only
Links: Not linkable

☛ Interpolated GainLoss Coeff Table

Type: Table
Units: TIME VS. FLOW VS. FRACTION
Description: a table relating the day of the year, flow rate, and the gainloss coefficient
Information: Data must be entered in blocks of increasing flow for each given day for the interpolation method to work correctly. January 1 is represented by a 1, February 1 as 32, etc. An example table is shown below.

Day of Year	Flow Rate	GainLoss Coeff
1	500	.01
1	550	.02
1	600	.03
32	500	.025
32	550	.03
32	600	.035
60	500	.03
60	550	.035
60	600	.04

I/O: Required input
Links: Not linkable

☛ Interpolated GainLoss Table

Type: Table
Units: TIME VS. FLOW VS. FLOW
Description: a table relating the day of the year, the flow rate and the gainloss value
Information: Data must be entered in blocks of increasing flow for each given day for the interpolation method to work correctly. January 1 is represented by a 1, February 1 as 32, etc. The table is set up in the same manner as the Interpolated GainLoss Coeff Table (shown above).
I/O: Required input
Links: Not linkable

☛ **TOTAL GAINLOSS**

Type: Series
Units: FLOW
Description: represents the total amount of flow that is gained or lost in a reach for a given timestep
Information: Applied to the bottom of the reach (it is added to the outflow value calculated from the Routing method).
I/O: Output only
Links: Not linkable

A 3D interpolation method is used to calculate a value for Interpolated GainLoss and Interpolated GainLoss Coeff using the Interpolated GainLoss Table and Interpolated GainLoss Coeff Table, respectively. Once these values are calculated Total GainLoss is set as:

$$\text{Total GainLoss} = \text{Interpolated GainLoss Coeff} \bullet \text{Routed Flow} + \text{Interpolated GainLoss}$$

22.1.10.6 Base Plus Fractional Loss

This method is used to model loss in a reach; it does not allow gain. Loss is calculated as base flow rate plus a proportion of the flow above the base. Loss is limited to be less than a maximum value. This method will be available for all Routing methods except Time Lag routing.

SLOTS ADDED BY THIS METHOD:

☛ **BASE LOSS**

Type: Periodic
Units: TIME VS FLOW
Description: Represents an amount of flow that is lost regardless of the flow rate in the reach. The method will limit the total loss to be less than or equal to the flow in the reach.
Information: A positive number represents a loss. A negative number results in an error.
I/O: Required Input
Links: Not linkable

☛ **LOSS FRACTION ABOVE BASE**

Type: Periodic
Units: TIME VS DECIMAL
Description: This slot represents a fraction of the flow above the Base Loss that will be lost
Information: A positive number represents a loss; the value must be between 0 and 1 or an error is issued.
I/O: Required Input
Links: Not Linkable

☛ Maximum Loss

Type: Scalar
Units: FLOW
Description: The maximum loss that can occur from the reach
Information: If not input, no maximum is used
I/O: Optional Input
Links: Not Linkable

☛ Total GainLoss

Type: Series
Units: FLOW
Description: Represents the total amount of flow that is gained or lost in a reach for a given timestep
Information: A positive number represents a gain, a negative number represents a loss. The Total GainLoss is applied to the bottom of the reach (it is added to the outflow value calculated from the Routing method). In this method, a loss is represented by a negative number.
I/O: Output only
Links: Not linkable

Method Details

At the beginning of the method, the values for **Base Loss** and the **Loss Fraction above Base** are taken from the periodic slots and checked. As is typical with periodic slots, these slots can be configured to either lookup or interpolate. If either is invalid, the run aborts with an error. If the **Base Loss** is less than zero, the run aborts with an error. If the **Loss Fraction above Base** is less than 0 or greater than 1, the run aborts with an error.

On a reach, the **Total GainLoss** slot is used by other GainLoss methods. As a result, this method follows the same convention where a negative **Total GainLoss** represents a loss. Also, the flow used by the method is the flow in the reach after the routing has occurred. Thus, the **Total GainLoss** is the base loss plus the fractional loss associated with the flow above the base. Because the flow could be less than the base, the following formula is necessary to calculate the loss:

$$tempLoss = min(BaseLoss, RoutedFlow) + max(RoutedFlow - BaseLoss, 0cms) \times (LossFractionAboveBase)$$

If **Maximum Loss** is NaN, then the **Total GainLoss** is set equal to the negative of the tempLoss value.

$$TotalGainLoss = -tempLoss$$

Otherwise, there is a valid **Maximum Loss** and the **Total GainLoss** is then set as:

$$TotalGainLoss = -min(tempLoss, MaximumLoss)$$

22.1.10.7 Periodic Gain Loss

This method is used to model gain and/or loss in a reach. The gain and loss values are entered as periodic data for a given time range. Gains are entered as positive values and losses are entered as negative values. This method will be available for all Routing methods except Time Lag routing.

SLOTS ADDED BY THIS METHOD:

☛ PERIODIC GAINLOSS

Type: Periodic
Units: TIME VS FLOW
Description: Represents an amount of flow that is gained or lost regardless of the flow rate in the reach.
Information: A positive number represents a gain. A negative number results in a loss.
I/O: Required Input
Links: Not linkable

☛ TOTAL GAINLOSS

Type: Series
Units: FLOW
Description: Represents the total amount of flow that is gained or lost in a reach for a given timestep
Information: A positive number represents a gain, a negative number represents a loss. The Total GainLoss is applied to the bottom of the reach (it is added to the outflow value calculated from the Routing method). In this method, a loss is represented by a negative number.
I/O: Output only
Links: Not linkable

METHOD DETAILS

At the beginning of the method, the value for **Periodic GainLoss** is taken from the periodic slot and checked. As is typical with periodic slots, this slot can be configured to either lookup or interpolate. If the value is invalid, the run aborts with an error.

This method follows the same convention as other GainLoss methods where a positive Total GainLoss represents a gain.

$$TotalGainLoss = Valuefrom(PeriodicGainLoss)$$

22.1.11 Reach Bank Storage

The Reach Bank Storage category is used to model flow to/from storage within a reach. The Bank Storage Return is added/removed from the reach after the flow is routed but before Gain Loss is calculated.

The Reach Bank Storage category is available for all Routing methods except time lag methods.

22.1.11.1 None

This is the default method. It performs no bank storage calculations. There are no slots specifically associated with this method.

22.1.11.2 Average Flow Bank Storage

This method is designed to simulate hyporheic zone storage and return on Reach objects. It is similar to the bank storage on reservoir objects. The amount of bank storage is calculated as a function of flow rate.

Slots Specific to this Method

☛ Average Flow Coefficient

Type: Table
Units: No Units
Description: A coefficient based on transmissivity of the surrounding aquifer and the relationship between river stage and flow rate
Information: Although the slot does not have a formal unit, the value in the slot has conceptual units of “square root of seconds”. Often this value is negative based on the equation below.
I/O: Required input
Links: Not linkable

☛ Bank Storage Return

Type: Series
Units: FLOW
Description: The rate of return to the reach from bank storage
Information: A positive number represents a return to the reach while a negative number represents a loss to bank storage.
I/O: Output only
Links: Not linkable

☛ Routed Flow

Type: Series
Units: Flow
Description: Flow rate used to calculate Bank Storage Return
Information: This represents the flow rate in the reach after Local Inflow, Seepage, and Diversion are added/removed and the flow is routed.
I/O: Output only
Links: Not linkable

☛ Timesteps to Average

Type: Table
Units: No Units
Description: An integer number of timesteps to compute the average flow. This value must be greater than zero
I/O: Required input
Links: Not linkable

$$\text{Bank Storage Return} = \frac{\text{Average Flow Coefficient} \times (\text{Routed Flow}_t - \text{routed flow average})}{\sqrt{\text{timestep length}}}$$

The routed flow average is the average flow over the number of timesteps specified in **Timesteps to Average** prior to the current timestep. For the first (timesteps to average - 1) timesteps, some routed flows are being set on timesteps prior to the initial timestep. These values, set equal to the routed flow at the initial timestep, are used so that calculations for routed flow average don't used routed flows equal to zero. A warning message will remind users this is occurring.

22.1.12 Reach Seepage

The Reach Seepage category is used to model seepage from a reach. The seepage is calculated at the top (upstream) end of the reach. All methods allow seepage to be input or set by a rule. When input or set by a rule, seepage is not recalculated by the method.

This method is available only when No Routing is selected in the Routing category.

22.1.12.1 None

This is the default method. It performs no seepage calculations. There are no slots specifically associated with this method.

22.1.12.2 Proportional Seepage

Seepage is calculated as a fraction of the Inflow to the reach.

Slots Specific to this Method

☛ **Maximum Seepage**

Type: Table
Units: FLOW
Description: represents the maximum amount of flow that can be lost as seepage
Information:
I/O: Required input
Links: Not linkable

☛ **Seepage**

Type: Series
Units: FLOW
Description: The calculated amount of flow lost as seepage
Information: Subtracted from the Inflow value to calculate Outflow.
I/O: Input, Output, or Rule
Links: May be linked to the Inflow slot on the Groundwater Storage object.

☛ **Seepage Flow Fraction**

Type: Table
Units: NO UNITS
Description: Represents the fraction of Inflow that is lost as seepage
Information: A decimal value which is greater than or equal to zero and less than one.
I/O: Required input
Links: Not linkable

$$Seepage = MIN(MaximumSeepage, Inflow \times SeepageFlowFraction)$$

22.1.12.3 Variable Seepage

This method uses variable parameters to model seepage. Seepage is calculated as a fraction of the Inflow for each time step.

Slots Specific to this Method

☛ Maximum Seepage

Type: Table
Units: FLOW
Description: represents the maximum amount of flow that can be lost as seepage
Information:
I/O: Required input
Links: Not linkable

☛ Seepage

Type: Series
Units: FLOW
Description: the calculated amount of flow lost as seepage
Information: Subtracted from the Inflow value to calculate Outflow.
I/O: Input, Output, or Rule
Links: May be linked to the Inflow slot on the Groundwater Storage object.

☛ Variable Seepage Flow Fraction

Type: Series
Units: DECIMAL
Description: a time series of values representing the fraction of Inflow that is lost as seepage
Information: Entered as a decimal value which is greater than or equal to zero but less than one.
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

$$Seepage = MIN(MaximumSeepage, Inflow \times VariableSeepageFlowFraction)$$

22.1.12.4 Horizontal Gradient Seepage

Each Reach may have horizontal seepage to multiple drains or shallow groundwater aquifers with unique drainage characteristics. Horizontal seepage to individual drains is calculated using Darcy's Law:

$$\text{Seepage} = \text{Hydraulic Conductivity} \times \text{Seepage Gradient} \times \text{Reach Bank Seepage Area}$$

The Seepage slot is set as the sum of horizontal seepage to all drains. An Aggregate Reach object can be used to simulate discrete reaches or reach segments for which varying seepage characteristics are known.

Slots Specific to the Method

☛ Seepage

Type: Series
Units: FLOW
Description: the calculated amount of flow lost as seepage
Information: Seepage is subtracted from the Inflow value to calculate Outflow.
I/O: Input, Output, or Rule
Links: May be linked to the Inflow slot on the Groundwater Storage object. This should only be done when groundwater is the only drain to which seepage flows.

☛ Stage vs Horiz Bank Seepage Area

Type: Table
Units: FLOW VS. AREA
Description: relates the reach avg stage to the horizontal bank area through which seepage flows in the vertical direction to each drain
Information: The user can add additional columns when more than one drain is present. If the bank seepage area is constant, two extreme values for stage with the same bank seepage area will insure that tale interpolations return this value.
I/O: Required input
Links: Not linkable

☛ Horiz Hydraulic Conductivity

Type: Table
Units: VELOCITY
Description: the horizontal hydraulic conductivity of the medium through which seepage flows
Information: The user can add additional columns when more than one drain is present.
I/O: Required input
Links: Not linkable

☛ Distance to Drain

Type: Table
Units: LENGTH
Description: the horizontal distance from the reach to the drain.
Information: This value is assumed to be constant. The user can add additional columns when more than one drain is present.
I/O: Required input
Links: Not linkable

The Horizontal Gradient Seepage method requires that a Stage method be selected. It also has a dependent method category, the Drain Elevation method category. The reach horizontal bank seepage area through which seepage can flow to each drain in the horizontal direction is calculated using a stage/bank seepage area table interpolation. The horizontal hydraulic conductivity of the medium adjacent to each drain is entered by the user. The horizontal seepage gradient for each drain is calculated using the equation:

$$\text{Horizontal Seepage Gradient} = \frac{\text{Avg Stage} - \text{Drain Elevation}}{\text{Distance to Drain}}$$

22.1.12.5 Vertical Gradient Seepage

Vertical gradient seepage is also calculated using Darcy's Law (see Horizontal Gradient Seepage Method).

Slots Specific to the Method

☛ Seepage

Type: Series
Units: FLOW
Description: the calculated amount of flow lost as seepage
Information: Seepage is subtracted from the Inflow value to calculate Outflow.
I/O: Input, Output, or Rule
Links: May be linked to the Inflow slot on the Groundwater Storage object if groundwater is the only drain to which seepage flows.

☛ Vert Hydraulic Conductivity

Type: Table
Units: VELOCITY
Description: the vertical hydraulic conductivity of the medium through which seepage flows
Information:
I/O: Required input
Links: Not linkable

☛ Vert Hydraulic Gradient

Type: Series
Units: NO UNITS
Description: the gradient through which seepage flows to the vertical drain
Information:
I/O: Required input
Links: Not linkable

☛ **VERT CHANNEL SURFACE AREA**

Type: Series
Units: AREA
Description: the surface area of the reach channel bottom through which seepage may flow in the vertical direction
Information:
I/O: Required input
Links: Not linkable

22.1.12.6 Horizontal and Vertical Gradient Seepage

The Horizontal and Vertical Gradient Seepage Method is a composite of the Horizontal Gradient Seepage and Vertical Gradient Seepage methods. When it is selected all of the slots for the Horizontal Gradient and Vertical Gradient Seepage methods (above) are instantiated. The Horizontal and Vertical Gradient Seepage method sums horizontal and vertical seepages.

22.1.12.7 Seepage and Riparian Consumptive Use Loss

The Seepage and Riparian CU Loss Method is only available when “Specify Local Inflow, Solve Outflow” or “No Local Inflow, Solve Outflow” is selected from the Local Inflow and Solution Direction category. The Seepage and Riparian CU Loss Method sets the Leakage slot either from the user, as input or by a rule, or calculated from Seepage and Riparian CU Loss with the following calculation:

$$\text{Leakage} = n_1 \times \log(\text{Inflow}) + n_2 \times (\text{Inflow})^2 + n_3 \times \text{Inflow} + n_4 \times \text{Riparian CU} + n_5 \times \text{Parallel Channel Flow}$$

where n_1, n_2, n_3, n_4, and n_5 are coefficients held in a table slot. For the Leakage calculation, if the inflow value is negative, it will be assumed to be zero.

The Seepage slot is set to the result of subtracting Riparian CU from the Leakage slot.

$$\text{Seepage} = \text{Leakage} - \text{Riparian CU}$$

The Riparian CU Loss slot is set to the minimum of the Riparian CU and the Leakage. Here, if the calculated Leakage is negative it is assumed to be zero.

$$\text{Riparian CU Loss} = \text{Min}(\text{Leakage}, \text{Riparian CU})$$

Note: The Leakage calculation uses the Inflow value in ***user units***. The value set in the LogInflowCoeff column and the value *and units* in the InflowSqrdCoeff column of the Evaporation Data slot (n1 and n2), as set by the user, must be consistent with the ***user units*** for the Inflow slot. Changing the user units on the Inflow slot will produce a different result. There will not be an automatic conversion of the Coefficient values.

SLOTS SPECIFIC TO THE METHOD

☛ SEEPAGE

Type: Series
Units: FLOW
Description: difference of Leakage and Riparian CU
Information: Subtracted from the Reach Inflow value to calculate Outflow.
I/O: Input, Output, or Rule
Links: Linkable

☛ LEAKAGE

Type: Series
Units: FLOW
Description: amount of incoming water lost to various sources.
I/O: Input or Output.
Links: Linkable

☛ RIPARIAN CU

Type: Series
Units: FLOW
Description: flow rate of the consumptive use in the riparian area
Information: Riparian CU is used in the Leakage calculation of the Reach and the Riparian CU Loss calculations.
I/O: Required Input
Links: Not linkable

☛ RIPARIAN CU LOSS

Type: Series
Units: FLOW
Description: the minimum of Leakage and Riparian CU
Information: Subtracted from the Reach Inflow value to calculate Outflow
I/O: Output only
Links: Not linkable

☛ SEEPAGE AND RIPARIAN CU LOSS COEFF

Type: Table
Units: NONE, 1/FLOW, NONE, NONE, NONE
Description: relates the month to a set of five coefficients with the following names and units: LogInflowCoeff [NONE], InflowSqrdCoeff[1/FLOW], InflowCoeff [NONE], CUCoeff [NONE], ParChanFlowCoeff[NONE]. the table has twelve rows, one for each month of the year.
Information: These coefficients are specific to the encoded Leakage equation. The values must be consistent with the user units for the Inflow slot. If the user units on

the Inflow slot are changed, the values in this slot must be changed manually to correspond to the new units. Otherwise different results will be produced

I/O: Required Input
Links: Not linkable

☛ Parallel Channel Flow

Type: Series
Units: Flow
Description: flow rate of upstream diversion to a parallel channel
Information: Available on Reach objects for use in Leakage calculations. Typically linked to flow in a parallel channel. Dispatch slot.
I/O: May be output or input. If it is not linked or user input, it will be assumed to be zero.
Links: Linkable

22.1.12.8 Head Based Seepage

A method to compute the river stage based on the reach inflow and outflow values must be selected in the Stage Calc Category. If no stage method is selected, the Head Based Seepage method will abort with an error. The Head Based Seepage method will compute the seepage based on the head in the river relative to the groundwater head. If the groundwater head is below the streambed elevation then the seepage will be computed based on the head in the river relative to the streambed bottom.

☛ Seepage

Type: Series Slot
Units: FLOW
Description: The river loss or gain from groundwater
Information: A positive number represents a losing reach while a negative number represents a gaining reach
I/O: Input, Output, or Rule
Links: Usually linked to the groundwater object

☛ Previous Water Table Elevation

Type: Series Slot
Units: LENGTH
Description: The previous elevation value computed by the connected groundwater object
Information: Seepage is computed based on the difference between the current river stage and the previous groundwater elevation
I/O: Output only
Links: Linked to the Previous Elevation slot on the connected groundwater object

☛ CONDUCTANCE

Type: Scalar slot
Units: AREA PER TIME
Description: The riverbed conductance
Information: The conductance is defined as the hydraulic conductivity of the streambed material, multiplied by the width of the streambed bottom, multiplied by the length of the stream segment, divided by the streambed thickness (Conductance = KwL/m).
I/O: Input or computed as specified HERE (Section 22.1.14).
Links: Not linkable

☛ STREAMBED ELEVATION

Type: Scalar Slot
Units: LENGTH
Description: The elevation of the stream bed at the center of the reach segment (i.e. average stream bed elevation)
Information: Should have the same datum as Stage and the Previous Water Table Elevation
I/O: Required input
Links: Not linkable

☛ CONDUCTANCE FACTOR

Type: Series Slot with Periodic Input
Units: FRACTION
Description: The Conductance is multiplied by this factor in the calculation Seepage. It can be used to model variable Conductance.
Information: If the slot is NaN, a default of 1 is used in the calculation of Seepage. The slot remain as NaN.
I/O: Optional input or rules
Links: Not typically linked

The Head Based Seepage Method will compute the seepage to groundwater as follows:

If the Previous Water Table Elevation is greater than the Streambed Elevation:

$$\text{Seepage} = \text{Conductance} \times \text{Conductance Factor} \times (\text{Stage} - \text{Previous Water Table Elevation}) \quad \textbf{Eq. 2}$$

If the water table elevation at the previous timestep is below the Streambed Elevation:

$$\text{Seepage} = \text{Conductance} \times \text{Conductance Factor} \times (\text{Stage} - \text{Streambed Elevation}) \quad \textbf{Eq. 3}$$

Note that the equations above use the water table elevation at the previous timestep. This simplification is done to avoid iteration problems that would result if the current groundwater elevation was used. A positive seepage is limited to be less than or equal to the Inflow to the reach.

22.1.13 Seepage Routing

The option to route seepage becomes visible after a method to calculate seepage is chosen.

22.1.13.1 No Routing

This is the default method which will not route seepage. The values for seepage are calculated depending on the user method chosen for seepage calculation. These values are contained in the Seepage slot.

22.1.13.2 Impulse Response

This method calculates the routed seepage based on the impulse response method of routing. This routing method makes visible a PreRouted seepage slot. This slot contains the values for seepage that have not been routed. The routed values are contained in the Seepage slot. If a link exists that links seepage to another object and a seepage routing method is chosen, the link will propagate the values of the routed seepage.

Note: This method computes the Seepage as a function of current and previous PreRouted Seepage. Because it uses previous timestep information, the computed Seepage may be more than the Inflow (or Outflow) and could lead to a negative Outflow (or Inflow). As always, Outflow = Inflow - Seepage + Sources - Sinks

SLOTS SPECIFIC TO THIS METHOD

☛ **NUMBER OF COEFFICIENTS**

Type: Table
Units: NONE
Description: The integer number of lag coefficients to be used in the method.
I/O: Required input
Links: Not linkable

☛ **LAG COEFFICIENTS**

Type: Table
Units: NONE
Description: The impulse response lag coefficients
Information: The number of lag coefficients must be the same as the value in the Number of Coefficients slot. The input will be in rows.
I/O: Required input.
Links: Not linkable.

☛ **PREROUTED SEEPAGE**

Type: Series
Units: FLOW
Description: The seepage values before routing.
Information: This slot contains the values of seepage that have not been routed. The routed values will be contained in the Seepage slot.
I/O: Output
Links: Can be linked, but slot containing routed values is Seepage. If routed values are to be linked, then this slot should not be linked.

The Seepage (routed) will be calculated as:

$$\text{Seepage}_t = C_0\text{PreRoutedSeepage}_t + C_1\text{PreRoutedSeepage}_{t-1} + \ldots.$$
$$+ C_{ncoeff-1}\text{PreRoutedSeepage}_{t-ncoeff-1} + C_{ncoeff}\text{PreRoutedSeepage}_{t-ncoeff}$$

where *C* is a lag coefficient and *ncoeff* is the number of lag coefficients.

22.1.13.3 One Timestep Seepage Lag

This method will apply the computed seepage to the next timestep.

SLOTS SPECIFIC TO THIS METHOD

☛ **SEEPAGE FROM PREVIOUS TIMESTEP CONDITIONS**

Type: Series
Units: FLOW
Description: The seepage computed from the previous timestep's conditions
Information:
I/O: Output Only
Links: Not linkable

Seepage is computed at each timestep, but the value is set at the next timestep.

$$\text{Seepage}_{t+1} = \text{computed seepage}_t$$

When this method is selected, the timing of the seepage calculation is modified. For the solveNRInflow and solveNROutflow dispatch methods, the Seepage Routing method is called once near the beginning of the dispatch method. This call makes sure that there is a valid seepage that was computed at the previous timestep. Note, if it is the first timestep in the run and there is no valid seepage, the method will use the current estimate of seepage and post a warning. (If you do not want to see this warning, input a seepage value for the first timestep.) When the Stage Table Lookup and Head Based Seepage are used, this estimate is the seepage based solely on the Inflow (for solveNROutflow) or Outflow (for solveNRInflow) to the reach.

When used with the Head Based Seepage method and solveNROutflow, a positive seepage is limited to be less than or equal to the inflow to the reach. The slot 'Seepage from Previous Timestep Conditions' holds the lagged seepage. If the reach redispatches with a larger inflow, the seepage is reset to this value. This reverts the seepage to its original value instead of using the seepage constrained by a smaller inflow.

The selected Reach Seepage method is called near the end of the dispatch method after inflow and outflow are known. For the Stage Table Lookup and Head Based Seepage, this execution allows the seepage to be correctly computed based on both the inflow and outflow. The seepage is then set at the NEXT timestep.

Note: Seepage is a dispatch slot, so the reach may re-dispatch at the next timestep, depending on knowns and unknowns.

For the solveNRLocalInflow dispatch method, the seepage calculation is only called once near the beginning of the dispatch method as both inflow and outflow are known. Again, on the first timestep, if a valid seepage is not known, the estimate based on current values will be used.

22.1.13.4 Step Response

This method calculates the routed seepage based on the step response method of routing. The Seepage slot contains the values for seepage that have not been routed. The routed values are contained in the Routed Seepage slot. Typically, the Routed Seepage slot will be linked.

Note: This method removes the Seepage from the reach and then routes the Seepage. The routed seepage is stored in the Routed Seepage slot for the current and n future timesteps. As always, Outflow = Inflow - Seepage + Sources - Sinks

This seepage routing method is available for all seepage methods except Head Based Seepage.

Slots Specific to this Method

☛ Lag Coefficients

Type: Table
Units: NONE
Description: The step response lag coefficients
Information: The input will be in rows.
I/O: Required input.
Links: Not linkable.

☛ Routed Seepage

Type: Agg Series Slot
Units: FLOW
Description: the seepage after it has been routed
Information:
I/O: Output Only
Links: Linkable

☛ Future Seepage Timesteps

Type: Scalar
Units: NONE
Description: The number of timesteps in the future (i.e. past the timestep at which the object is dispatching) at which Routed Seepage will be computed and set.
Information: This slot must be an input integer and greater than or equal to 1. If you do not want it to set future timesteps, input a value of 1.
I/O: Required Input
Information: Not Linkable

This method sets values into the future as follows:

nCoeff = Number of rows in the Lag Coefficients table

nFutureTimesteps = Min (Future Seepage Timesteps, nCoeff)

For each timestep k from (t to (t + nFutureTimesteps - 1))

$$\text{Routed Seepage}_k = C_0\text{Seepage}_k + C_1\text{Seepage}_{k-1} + \ldots. + C_{nCoeff-1}\text{Seepage}_{k-ncoeff-1} + C_{nCoeff}\text{Seeepage}_{k-nCoeff}$$

End for

Thus, the method computes and sets Routed Seepage forward nFutureTimesteps.

When the method is looking for a Seepage for a timestep that is past the dispatching timestep and the value is not valid, the Seepage is assumed to be zero. If the Seepage is not valid for a previous timestep, the calculation will exit the method and post a warning message. No Routed Seepage will be set.

22.1.14 Reach Conductance

Methods in this category allow you to choose how you wish to specify conductance. You can either give values for Conductance or give values for Hydraulic Conductivity and the geometry of the reach and the Conductance will be computed.

22.1.14.1 Specify Conductance

This is the default method and does not instantiate any new slots. You must specify the conductance value or an error will be issued.

22.1.14.2 Compute Conductance

This method allows you to specify hydraulic conductivity and the geometry.

THE FOLLOWING SLOTS ARE ADDED:

☛ **HYDRAULIC CONDUCTIVITY**

Type: Scalar
Units: VELOCITY
Description: Hydraulic conductivity of the streambed material
Information:
I/O: Required Input
Links: Not Linkable

☛ **RIVERBED THICKNESS**

Type: Scalar
Units: LENGTH
Description: Thickness of the riverbed
Information: This value is used in the conductance calculation
I/O: Required Input
Links: Not Linkable

☛ **SEEPAGE AREA**

Type: Scalar
Units: AREA
Description: Area of the river that contributes to seepage
Information:
I/O: Required Input
Links: Not Linkable

At the start of the run, the information in these slots is used to compute the **Conductance** value according to the following equation:

$$\text{Conductance} = \text{Hydraulic Conductivity} \times \frac{\text{Seepage Area}}{\text{Riverbed Thickness}}$$

If there are missing values in the above slots, an error will be issued and the run initialization will be stopped.

The resulting value is set on the scalar **Conductance** slot. If there are values already in the slot, they will be overwritten.

Note: The **Conductance** slot is registered as having a “Source” slot. When a slot has a source slot, it becomes read-only and displays a cross hatch over the data. It also provides a note indicating the source used to compute the data. This attribute is set at the start of the run, so you must initialize the model to see the display of this slot change. If you deselect this method, you must initialize the run again to un-set the source slot.

22.1.15 Drain Elevation

The Drain Elevation category is dependent on the Horizontal and Horizontal and Vertical Gradient Seepage Methods. Each Reach may have horizontal seepage to multiple drains or shallow groundwater aquifers, each of which can have unique drainage characteristics. Drain related slots are designed so that users can add columns to provide information on each drain present. A “drain” may be a man-made drain or groundwater.

22.1.15.1 None

None is the default for the category, and will result in an error if it is selected and a run is started.

22.1.15.2 Constant

This method allows the user to set a constant drain elevation.

Slots Specific to the Method

☛ **Constant Drain WS Elev**

Type: Table
Units: Length
Description: the elevation of the drain
Information: The user may add additional columns if more than one drain is present.
I/O: Required input
Links: Not linkable

22.1.15.3 Variable Input

This method allows the user to specify the drain elevation at each timestep.

Slots Specific to this Method

☛ **Variable Drain WS Elev**

Type: AggSeries
Units: Length
Description: the drain elevation at each timestep
Information: The user may add additional columns if more than one drain is present.
I/O: Required input
Links: Not linkable

22.1.15.4 Flow Elevation Interpolation

This method calculates the drain elevation based upon flow through the drain (this method should be used only when drains are man-made as flow is unavailable for groundwater). A table interpolation of the Drain Flow vs Drain Elev table gives the elevation. The slots for this method are:

☛ **DRAIN FLOW**

Type: Agg Series
Units: FLOW
Description: the flow through the drain
Information: The user may add additional columns if more than one drain is present.
I/O: Required input
Links: Not linkable

☛ **DRAIN FLOW VS DRAIN ELEV**

Type: Table
Units: FLOW VS. LENGTH
Description: a table relating the drain flow to the drain elevation
Information: The user may add additional columns if more than one drain is present.
I/O: Required Input
Links: Not linkable

22.1.15.5 Stage Elevation Interpolation

This method should be used when the drain or groundwater elevation depends on the Reach Avg Stage (water surface elevation). It calculates the drain elevation based upon a specified table relationship of Reach Avg Stage and Drain Elev. The user may also include a time lag using the Drain Time Lag slot representing a lagged relationship between the Reach and drain elevations.

☛ **REACH STAGE VS DRAIN ELEV**

Type: Table
Units: LENGTH VS. LENGTH
Description: table relating the reach average stage (water surface elevation) to the drain elevation
Information: The user may add additional blocks of columns if more than one drain is present.
I/O: Required Input
Links: Not linkable

☛ **DRAIN TIME LAG**

Type: Table
Units: TIME
Description: the time for which the elevation should be lagged
Information: The user may add additional columns if more than one drain is present.
I/O: Input only
Links: Not linkable

22.1.16 Outflow Adjustment

This method adjusts outflows after they have been solved for by the dispatch method. This category is available for all Routing methods.

22.1.16.1 None

This method is the default for this category. If no adjustment is desired, this should be selected.

There are no slots specifically associated with this method.

22.1.16.2 Negative Outflow Unidentified Loss Adjust

The user should select this method when negative outflows need to be avoided. When using this method, negative outflows, as calculated by the Routing method, will be reset to zero.

Slots Specific to this Method

☛ **Unidentified Loss**

Type: Series
Units: FLOW
Description: a value that give the absolute value of the negative outflow that was set to zero.
Information:
I/O: Output.
Links: Not Linkable.

22.1.16.3 Autoregressive Outflow

The user should select this method if they want to adjust the current outflow to a weighted sum of the current routed flow and prior outflows. It is available for use with all Routing methods except Step Response and Variable Step Response, but it is only available for Time Lag and No Routing if the selected method in the Local Inflow and Solution Direction category is “Specify Local Inflow, Solve Outflow” or “No Local Inflow, Solve Outflow”. That is, this method can only be used when solving downstream.

Slots Specific to this Method

☛ Routed Flow

Type: Series
Units: FLOW
Description: This is the routed flow before the autoregression is applied, set as an output on the slot so the user can see its value.
Information: These values are computed by the Routing method and includes all gains and loss terms on the reach
I/O: Output only.
Links: N/A.

☛ Autoregressive Flow Coefficients

Type: Table
Units: No Units
Description: The coefficients that will be applied in the autoregression.
Information: The number of rows indicate the number of timesteps to use. The value in the first row is applied to the current Routed Flow. The value in the second row is applied to the first autoregressive term, i.e. Outflow[t-1]. Subsequent coefficients are applied to additional autoregressive terms, i.e. Outflow[t-(row-1)]. These coefficients should sum to 1.0 or an error will be issued at start of run. Use the **View -> Show Column Sum Row** to verify the values sum to 1.0. Most likely, there should be two or more rows.
I/O: Required Input.
Links: N/A

The Outflow is computed as follows:

$$\begin{aligned}\text{Reach.Outflow}[t] = & B_1 \times \text{Reach.Routed Flow}[t] \\ & + B_2 \times \text{Reach.Outflow}[t-1] \\ & + \ldots \\ & + B_N \times \text{Reach.Outflow}[t-N-1]\end{aligned}$$

Where B_1 through B_N are the coefficients in the **Autoregressive Flow Coefficients** slot. If the Routed Flow or Outflow is not valid, an error will be issued. At the beginning of the run, some pre-run Outflows may need to be input for the Routing method to compute the first Routed Flow. If previous outflows are not available, but the timestep is within the first few timesteps of the run (defined as number of autoregressive coefficients plus offset/lag determined by the Routing method), the outflow will be set to the Routed Flow.

22.1.17 Diversion from Reach

This method category controls how the reach handles diversion. This category is available for all Routing methods except Step Response, Time Lag, Storage Routing, and Variable Storage Routing.

22.1.17.1 None

This method is the default for this category. If no diversion is desired, this should be selected.

There are no slots specifically associated with this method.

22.1.17.2 Available Flow Based Diversion

The user should select this method when an Agg Diversion Site, WaterUser, or a Diversion Object (using Available Flow Diversion) is linked to the Reach. Diversion could also be set as an input.

Slots Specific to this Method

☛ **Available for Diversion**

Type: Series
Units: FLOW
Description: holds divertable flow in the reach, usually the inflow.
Information:
I/O: Usually output, but can be input or set by a rule.
Links: Linked to the Available For Diversion slot on any object, the Incoming Available Water slot on a WaterUser, or the Total Available Water slot on an Agg Diversion Site.

☛ **Diversion**

Type: Series
Units: FLOW
Description: holds diversion value from Agg Diversion Site, or input.
Information:
I/O: Optional; set to 0.0 if not input and not linked.
Links: Usually linked to the Diversion slot on the Agg Diversion Site, WaterUser, or the Diversion Object.

☛ **DIVERSION CAPACITY**

Type: Scalar Slot
Units: FLOW
Description: used to hold the maximum diversion physically possible from the reach
Information: This slot is used in the accounting system for allocation purposes and can be used in Rulebased Simulation
I/O: Input only
Links: Not linkable

METHOD DETAILS:

If the Available for Diversion is specified (input or set by a rule) OR the linked slot is specified (input or set by a rule), no further computations are performed. The specified Available for Diversion is used in the dispatch method.

When solving downstream:

If a Min Diversion Bypass method is selected, Available for Diversion is calculated as Inflow minus Minimum Diversion Bypass. If there is a minimum specified on the Outflow (in the configuration dialog), the Available for Diversion is set to the Inflow minus the minimum outflow. Otherwise, Available for Diversion is set to the value in the Inflow slot. Available for Diversion is limited to be larger than zero, unless there is a negative inflow.

When solving upstream and Diversion is not valid but linked:

Available for Diversion is set to the maximum value specified on the configuration dialog for Available for Diversion. If no maximum is configured, an error will stop the run.

Note: Local Inflows into the reach are not included in the Available for Diversion.

22.1.17.3 Head Based Diversion

This method may be selected when a Diversion Object, using either the Gravity or Pumped Diversion method, is linked to the Reach. The “Stage Table Look Up” or “Inflow Stage Table Look Up” method must be selected if this method is selected.

This method allows the diversion object to check how much water is available to be diverted based on the elevation of the flow in the river.

SLOTS SPECIFIC TO THIS METHOD

☛ Diversion

Type: Series
Units: FLOW
Description: holds diversion value from Diversion Object, or input.
Information:
I/O: Optional; set to 0.0 if not input and not linked.
Links: Usually linked to the Diversion slot on the Agg Diversion Site, WaterUser, or the Diversion Object.

☛ Diversion Capacity

Type: Scalar Slot
Units: FLOW
Description: used to hold the maximum diversion physically possible from the reach
Information: This slot is used in the accounting system for allocation purposes and can be used in Rulebased Simulation
I/O: Input only
Links: Not linkable

☛ Water Elevation at Diversion

Type: Series
Units: FLOW
Description: holds water elevation at diversion
Information: This slot should be linked to the Diversion Object's Diversion Intake Elevation slot.
I/O: Output only
Links: Linked to the Diversion Intake Elevation

22.1.18 Min Diversion Bypass

The Min Diversion Bypass methods are only available when Available Flow Based Diversion is selected in the Diversion from Reach method category. It is used to specify the minimum amount of water that should remain in the reach after diversion. After the diversion, however, the reach may lose additional water to seepage or gain water from return flow.

22.1.18.1 None

This is the default method. It performs no calculations. There are no slots specifically associated with this method.

22.1.18.2 Input Min Bypass

This method is used when the user wants to input the minimum amount of water that bypasses the diversion point.

Slots Specific to this Method

☛ Minimum Diversion Bypass

Type: Series
Units: FLOW
Description: minimum amount of water remaining after diversion
Information:
I/O: Optional; not used if not input
Links: Not linkable

If Minimum Diversion Bypass is input, it used to calculate Available For Diversion as follows: $\text{Available For Diversion} = \text{Inflow} - \text{Minimum Diversion Bypass}$

22.1.18.3 Monthly Min Bypass Values

This method is used when the Minimum Diversion Bypass is dependent upon the month.

Slots Specific to this Method

☛ Minimum Bypass Values

Type: Table
Units: FLOW
Description: a table of minimum diversion bypass values for each month
Information: One value is required for each month.
I/O: Required input
Links: Not linkable

☛ **MINIMUM DIVERSION BYPASS**

Type: Series
Units: FLOW
Description: minimum amount of water remaining after diversion
Information: Calculated from the Minimum Bypass Values table
I/O: Output only
Links: Not linkable

Minimum Diversion Bypass is set with a value from the Minimum Bypass Values table for the current month in the simulation. Available For Diversion is then calculated as:

$$\text{Available For Diversion} = \text{Inflow} - \text{Minimum Diversion Bypass}$$

22.1.18.4 Monthly Min Bypass Coefficients

This method is used to compute Minimum Diversion Bypass from a coefficient that is dependent on the month.

SLOTS SPECIFIC TO THIS METHOD

☛ **MINIMUM BYPASS COEFFICIENTS**

Type: Table
Units: NONE
Description: a table of bypass coefficients for each month
Information:
I/O: Required input
Links: Not linkable

☛ **MINIMUM DIVERSION BYPASS**

Type: Series
Units: FLOW
Description: minimum amount of water remaining after diversion
Information:
I/O: Output only
Links: Not linkable

Minimum Diversion Bypass is calculated with a value from the Minimum Bypass Coefficients table for the current month in the simulation by the following equation:

$$\text{Minimum Diversion Bypass} = \text{Minimum Bypass Coefficients(current month)} \times \text{Inflow}$$

Available For Diversion is then calculated as:

$$\text{Available For Diversion} = \text{Inflow} - \text{Minimum Diversion Bypass}$$

22.1.18.5 Periodic Min Bypass

This method computed the Minimum Diversion Bypass as a function of date/season as defined in a periodic slot.

Slots Specific to this Method

☛ Periodic Minimum Bypass

Type: Periodic Slot
Units: FLOW
Description: The minimum diversion bypass as a function of season/date
Information: The current date is used to look up the Minimum Diversion Bypass from this slot.
I/O: Required Input
Links: Not linkable

☛ Minimum Diversion Bypass

Type: Series
Units: FLOW
Description: Minimum amount of water remaining after diversion
Information: This value is determined from the Periodic Minimum Bypass slot.
I/O: Output only
Links: Not linkable

22.1.19 Reach Evaporation

This method category is used to model evaporation on a reach object.

22.1.19.1 None

This is the default method. No calculations are performed and there are no slots associated with this method. It should be used if the user does not wish to model evaporation.

22.1.19.2 Inflow Exponent Pan Evaporation

This method models evaporation based on empirical equations.The equations are as follows:

$$\text{For Inflow} = Q < \text{Threshold Flow} \quad \text{Loss} = PanEvaporation(Coefficient \times Q^{Expon} + PanEvapCoeff(SurfaceArea - Coefficient \times Q^{Expon}))$$

$$\text{For Inflow} = Q >= \text{Threshold Flow} \quad \text{Loss} = PanEvaporation(Coeff \times Q^{Expon})$$

Note: The exponential calculation is carried out in ***user units***. The value set in the Exponent column of the Evaporation Data slot, as set by the user, must be consistent with the ***user units*** for the Inflow slot. Changing the user units on the Inflow slot will produce a different result. There will not be an automatic conversion of the Exponent value.

Slots Specific to this Method

☛ Evaporation Data

Type: Table
Units: FLOW, TIME/LENGTH, NONE, AREA, NONE
Description: This table holds various coefficients and constants for the **Inflow Exponent Pan Evaporation** method. It holds the following values and units: Threshold Flow (FLOW), Coefficient (TIME/LENGTH), Exponent (NONE), Bankfull Surface Area (AREA), PanEvapCoeff (NONE).
Information: The value in the Exponent column must be consistent with the user units for the Inflow slot. If the user units on the Inflow slot are changed, the Exponent must be changed manually to correspond to the new units. Otherwise different results will be produced.
I/O: Input Only
Links: Not linkable

☛ **EVAPORATION**

Type: Series Slot
Units: FLOW
Description: This slot is the output to the **Inflow Exponent Pan Evaporation** method. It holds the losses as calculated by the empirical equations.
Information:
I/O: Output only
Links: Not linkable

☛ **PAN EVAPORATION**

Type: Series
Units: VELOCITY
Description: This slot contains the pan evaporation coefficient
Information: This slot must have a valid value for the Inflow Exponent Pan Evaporation method.
I/O: Input only
Links: Can be linked to data object

22.1.19.3 Pan Evaporation

This method models evaporation based on the pan evaporation rate, the surface area of the reach and a pan evaporation coefficient.

SLOTS SPECIFIC TO THIS METHOD

☛ **EVAPORATION**

Type: Series Slot
Units: FLOW
Description: This slot is the output to the Pan Evaporation method. It holds the losses as calculated by the equation.
Information:
I/O: Output only
Links: Not linkable

☛ **PAN COEFFICIENT**

Type: Periodic
Units: NOUNITS
Description: This slot holds a coefficient that is applied to the computation.
Information: The coefficient should be greater than 0.0.
I/O: Required Input
Links: NA

☛ INFLOW SURFACE AREA TABLE

Type: Table
Units: FLOW VS AREA
Description: Table relating Inflow to the reach to Surface Area
Information:
I/O: Input Only
Links: NA

☛ PAN EVAPORATION

Type: Series
Units: VELOCITY (LENGTH/TIME)
Description: This slot contains the pan evaporation rate.
Information: This slot must have a valid input or an error will occur.
Information: Input only
I/O: Can be linked

☛ SURFACE AREA

Type: Series
Units: AREA
Description: The water surface area of the reach.
Information: The value in this slot is found by looking up the **Inflow** on the **Inflow Surface Area Table**
I/O: Output only
Links: Linkable, possibly to the Groundwater Storage object “Wetted Sand Area Excluded” slot. See Section 14.1.11.2, “Wetted Sand Evaporation” for more information.

METHOD DETAILS

To calculate evaporation, first, the **Inflow** is looked up on the **Inflow Surface Area Table** to compute and set the **Surface Area**. Then, the **Pan Coefficient** and **Pan Evaporation** are accessed and verified that there is valid data. Finally, **Evaporation** is computed as follows:

$$Evaporation[t] = SurfaceArea[t] \times PanCoefficient[t] \times PanEvaporationRate[t]$$

22.1.20 MODFLOW Link Category Reach

The MODFLOW Link Category is dependent upon the selection of No Routing, Time Lag, Variable Time Lag, Muskingum Cunge, or Muskingum Cunge Improved methods.

Note: RiverWare's connection with MODFLOW is currently not functional. This category is hidden and the methods have been disabled and cannot be selected. An error will be posted at model load if this method was previously selected. Contact CADSWES for help.

22.1.20.1 No Link to MODFLOW

No computations or slots are associated with the No Link to MODFLOW method, this is the default method for the MODFLOW Link Category.

22.1.20.2 Link to MODFLOW

The Link to MODFLOW method will allow a Reach to be linked with MODFLOW. The Link to MODFLOW method does not preform any computations but is used to configure the model. Data transferred between RiverWare and MODFLOW, as well as any interpolation and summation of this data, will be handled within the computational subbasin framework. Note, RiverWare assumes a consistent datum throughout all related objects' elevation and stages slots. Click HERE (Section 7.2.1) to view the "RiverWare - MODFLOW Connection" Functionality Guide. A description of the Reach specific data configuration is presented in that guide HERE (Section 7.2.1.3.2).

Slots Specific to this Method

☛ Total MODFLOW GainLoss

Type: Series Slot
Units: FLOW
Description: The total gain or loss between the reach and the aquifer
Information: A positive number represents a gain while a negative number represents a loss, this value is applied to the bottom of the reach, i.e. it is added to the routed flow value calculated from the Routing method)
I/O: Output only
Links: Linkable

Flux between the reach and underlying groundwater will be calculated in MODFLOW. These individual MODFLOW cell fluxes will be summed within the computational subbasin framework, and the summed value will be automatically mapped to the Total MODFLOW GainLoss slot.

When using the Link to MODFLOW method, either the “Inflow Stage Table Look Up” or “Stage Table Look Up” method must be selected. A stage value is needed to calculate the Flux between the River and underlying groundwater and will be transferred through the computational subbasin framework to MODFLOW.

If desired, when both the Link to MODFLOW method and a Diversion from Reach method are selected the existing Diversion slot value may be transferred to surface water body represented in MODFLOW through the computational subbasin framework. Note: the Diversion from Reach Category is not available for the Time Lag routing method.

A return flow from a surface water body represented in MODFLOW to a RiverWare Reach may be desired. When the Link to MODFLOW method and the Local Inflow MODFLOW Return method are selected a surface water return flow may be transferred from MODFLOW to the RiverWare Local Inflow MODFLOW Return slot through the computational subbasin framework.

22.1.21 Alternative Routing on Subbasin

This method category is used to provide alternative routing coefficients for use in subbasin methods.

22.1.21.1 None

This is the default method. No calculations are performed and there are no slots associated with this method.

22.1.21.2 Variable Step Coefficients

This method is used to provide variable step coefficients for use in subbasin methods that aggregate routing coefficients for a control point. A description of the variable routing is described **HERE (USACE_SWD.pdf, Section 2.10)**. This method allows the reach to solve one way in simulation dispatching, but another way in these subbasin methods.

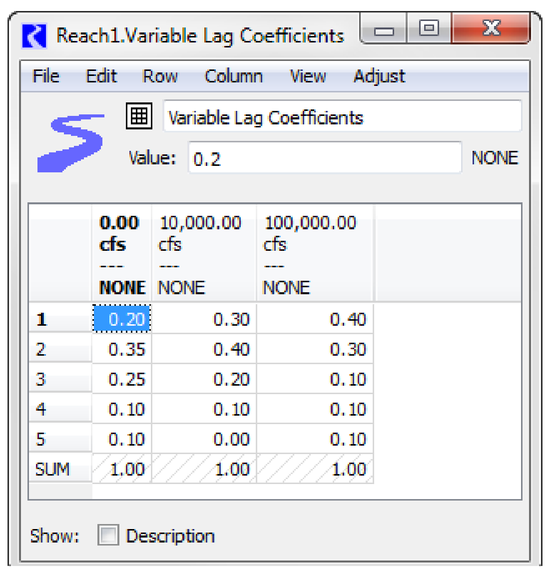
SLOTS ADDED BY THE METHOD

☛ **VARIABLE LAG COEFFICIENTS**

Type: Table Slot

Units: COLUMN MAP VALUES - FLOW, TABLE VALUES - NO UNITS

Description: A table defining the step response coefficients for each inflow threshold as shown in the following sample.



Information: This table has a Column Map which means that each column has an associated numerical value (with units). This numerical value is displayed on as the column label. Columns are added and deleted from this table using the **Column** menu with the following options: **Set Number of Columns**, **Append Column**, **Delete Column** and **Delete Last Column**. User units, scale, type, and precision for the Column Map (i.e. the column heading values) are defined in the unit scheme for Flow unit types. Column map values are set from the **Column, Set Column Value** menu option. When a column value is changed, the columns will re-order to ensure that the column values are increasing left to right. The sum of coefficients in each column should equal 1.0. This can easily be verified by adding a summary row at the bottom of the table using the **View**, **Show Column Sum Row** menu option. When used in the Variable Step Response method, the column map values are used as a stair step lookup, e.g. an Inflow of 500cfs uses the first column, an inflow of 10,000cfs uses the second column, and an inflow of 120,000cfs uses the third column. Therefore, the column map values need not bound the highest expected flows; flows greater than the largest column map value use the right most column. Note, the left-most set of coefficients should represent the minimum flow in the reach.

I/O: Required Input

Links: Not Linkable

22.2 Dispatch Methods

The dispatch methods for the Reach allow the various Routing methods to solve, and execute whenever the reach has enough information to solve.

22.2.1 SolveNROutflow

This method is available for the No Routing user method. It solves for Outflow when Inflow, and possibly Local Inflow, Return Flow, and Diversion are known, based on user method selection and the status of those slots.

Required Knowns

- **Inflow**
- **Local Inflow** (If one of the “Solve” or Contingent Local Inflow or Solve Outflow methods is selected, HERE (Section 22.1.3))
- **Local Inflow MODFLOW Return** (if the “Local Inflow MODFLOW Return” user method is selected)

Required Unknowns

- **Outflow**

If diversion or return flow is linked, the dispatch method will wait for a value in these slots before solving. If a value is set for the minimum outflow, the solved value for Outflow will be compared to the minimum. If the minimum has not been met, the outflow will be raised to the minimum, and the necessary inflow will be placed in the Local Inflow Adjust slot, if it is visible.

This dispatch method executes the methods in the following order:

1. Stage Calc
2. Reach Seepage (if One Timestep Seepage Lag is not selected)
3. Seepage Routing
4. Min Diversion Bypass
5. Diversion from Reach
6. Reach Evaporation
7. Reach Bank Storage
8. MODFLOW Link Category Reach

9. Gain Loss
10. mass balance:

$$\begin{aligned} Outflow = {} & Inflow \\ & + ReturnFlow \\ & - Diversion \\ & + LocalInflow \\ & + TotalGainLoss \\ & - Seepage \\ & + BankStorageReturn \\ & - Evaporation \\ & + LocalInflowMODFLOWreturn \\ & - TotalMODFLOWGainLoss \\ & + RiparianConsumptiveUseLoss \end{aligned}$$

11. Negative Outflow Adjustment
12. Stage Calc
13. Reach Seepage (if One Timestep Seepage Lag is selected)

22.2.2 SolveNRInflow

This method is available for the No Routing user method. It solves for Inflow when Outflow, and possibly Local Inflow, Return Flow, and Diversion are known, based on user method selection and the status of those slots.

Required Knowns

☛ **Outflow**

☛ **Local Inflow** (If the “Specify Local Inflow, Solve Inflow or Outflow” user method is selected)

☛ **Local Inflow MODFLOW Return** (if the “Local Inflow MODFLOW Return” user method is selected)

Required Unknowns

☛ **Inflow**

If diversion or return flow is linked, the dispatch method will wait for a value in these slots before solving. This dispatch method executes the methods in the following order:

1. Stage Calc
2. Diversion from Reach
3. MODFLOW Link Category Reach

4. Gain Loss
5. Reach Bank Storage
6. Reach Seepage (if One Timestep Seepage Lag is not selected)
7. Seepage Routing
8. mass balance:

$$\begin{aligned} Inflow = Outflow \\ & - ReturnFlow \\ & - LocalInflow \\ & + Diversion \\ & - TotalGainLoss \\ & - TotalMODFLOWGainLoss \\ & - LocalInflowMODFLOWReturn \\ & + Seepage \\ & - BankStorageReturn \end{aligned}$$

9. Stage Calc
10. Reach Seepage (if One Timestep Seepage Lag is selected)

22.2.3 SolveNRLocalInflow

This method is available for the No Routing user method. It solves for Local Inflow when Inflow and Outflow, and possibly Return Flow, and Diversion are known, based on user method selection and the status of those slots.

Required Knowns

- **Inflow**
- **Outflow**

Required Unknowns

- **Local Inflow**

If diversion or return flow is linked, the dispatch method will wait for a value in these slots before solving. This dispatch method executes the methods in the following order:

It first solves down

1. Stage Calc
2. Reach Seepage (if One Timestep Seepage Lag is not selected)
3. Seepage Routing
4. Reach Evaporation

5. Min Diversion Bypass
6. Diversion from Reach

It then solves up for the following:

7. MODFLOW Link Category Reach
8. Gain Loss
9. Reach Bank Storage

It then solves mass balance as:

$$\begin{aligned} LocalInflow = {} & Outflow - Inflow \\ & + Diversion \\ & - ReturnFlow \\ & - LocalInflowAdjust \\ & - TotalGainLoss \\ & - TotalMODFLOWGainLoss \\ & - BankStorageReturn \\ & + Seepage \\ & + Evaporation \\ & + RiparianConsumptiveUseLoss \end{aligned}$$

22.2.4 solveTLInflow

This dispatch method is available for the Time Lag routing user method. This is a front for the Time Lag, Stage Calc, and Volume Calc methods, which do all the work. Inflow is found at some previous time when Outflow is known at the current timestep.

REQUIRED KNOWNS

☛ **OUTFLOW**

REQUIRED UNKNOWNS

☛ **INFLOW**

22.2.5 solveTLOutflow

This dispatch method is available for the Time Lag routing user method. This is a front for the Time Lag, Stage Calc, and Volume Calc methods, which do all the work. Outflow is found at some future time when Inflow is known at the current timestep.

REQUIRED KNOWNS

☛ **INFLOW**

REQUIRED UNKNOWNS

☞ **OUTFLOW**

22.2.6 solveTLInflowOutflow

This dispatch method is available for the Time Lag routing user method. This is a front for the Time Lag, Stage Calc, and Volume Calc methods, which do all the work. Outflow is found at some future date, and inflow is found at some previous time, when both are known at the current timestep.

REQUIRED KNOWNS

☞ **INFLOW**

☞ **OUTFLOW**

22.2.7 solveTLOutflowDSOnly

This dispatch method is available if Time Lag routing is selected and "Specify Local Inflow, Solve Outflow" or "No Local Inflow, Solve Outflow" user methods are selected. Outflow is found at some future timestep when inflow is known on the current timestep. Unlike the solveTLOutflow dispatch method which is able to solve both upstream and downstream depending on the knowns and unknowns, solveTLOutflowDSOnly can only solve in the downstream direction. In addition, this dispatch method calls the Stage Calc and Volume Calc methods.

REQUIRED KNOWNS

☞ **INFLOW**

22.2.8 Solve Storage Routing Outflow

This dispatch method is available for the Storage Routing, Variable Storage Routing, and Modified Puls routing methods. The dispatch method executes the Routing, Stage Calc, and Volume Calc methods, which do all the work. Outflow is found at the current timestep when inflow is known at the current timestep.

REQUIRED KNOWNS

☞ **INFLOW**

REQUIRED UNKNOWNS

☞ **OUTFLOW**

22.2.9 solveOutflow

This is the dispatch method which all other Routing methods use. It executes the Stage, Min Diversion Bypass, Diversion from Reach, and then the Routing user methods. After the routing method, the Stage method is called again.

Required Knowns

☛ **Inflow**

Required Unknowns

☛ **Outflow**

Note, if the “Step Response”, “Variable Step Response”, or “Variable Time Lag” routing method is selected, Outflow is NOT a required unknown.

23. Slope Power Reservoir

Similar to the Level Power Reservoir, except Storage includes the "wedge" formed by a sloped water surface.

General Slots

☛ BACKWATER ELEVATION

Type: Series Slot
Units: LENGTH
Description: water surface elevation at the upstream end of a slope power reservoir
Information:
I/O: Optional; can be input or linked.
Links: Can be linked to the Tailwater Elevation or Tailwater Base Value slots of an upstream reservoir.

☛ CANAL FLOW

Type: Series Slot
Units: FLOW
Description: flow into (out of) the reservoir from (to) a canal
Information: May be linked to either the Flow 1 or Flow 2 slot of the Canal object. If not linked, the slot is set to zero.
I/O: Output only
Links: May be linked to either the Flow 1 or Flow 2 slot of the Canal object. If not linked, the slot is set to zero.

☛ CONVERGENCE PERCENTAGE

Type: Table
Units: NONE
Description: A percentage value ranging from 0 to 1 used for convergence in all iterative calculations
Information: Click **HERE (Appendix A: Reservoir Convergence)** for more information on the convergence algorithm
I/O: Optional; defaults to 0.0001 if not input.
Links: Not linkable

☛ Diversion

Type: SeriesSlot
Units: FLOW
Description: flow from the reservoir to a diverting object
Information: If not linked or input it is set to zero.
I/O: Optional; may be input or linked or neither
Links: May be linked to the Total Diversion slot on an Agg Diversion Site or the Total delivery Request slot on an AggDistribution Canal.

☛ Diversion Capacity

Type: Scalar Slot
Units: FLOW
Description: used to hold the maximum diversion physically possible from the reservoir
Information: This slot is used in the accounting system for allocation purposes and can be used in Rulebased Simulation
I/O: Input only
Links: Not linkable

☛ Elevation Volume Table

Type: Table
Units: LENGTH vs. VOLUME
Description: Reservoir Pool Elevation vs. Reservoir Storage
Information:
I/O: Required input
Links: Not linkable

☛ Energy

Type: Series Slot
Units: ENERGY
Description: product of the power generated by flow through the turbines and the length of the timestep
Information: This slot may also take the BEST_EFFICIENCY or MAX_CAPACITY flags, which allow Energy to act as input for dispatching, but solve for the value of energy assuming the generators are operating at best efficiency or maximum capacity.
I/O: Optional; if not input by the user, Energy is computed in the power calculations.
Links: Usually not linked

☛ Flow FROM Pumped Storage

Type: Series Slot

Units: FLOW
Description: flow into the reservoir from a pumped storage reservoir
Information: May be linked to the Outflow slot of a Pumped Storage object.
I/O: Optional; usually linked if used.
Links: May be linked to the Outflow slot of a Pumped Storage object.

☛ Flow TO Pumped Storage

Type: Series Slot
Units: FLOW
Description: flow out of the reservoir into a pumped storage reservoir
Information:
I/O: Optional; usually linked if used
Links: May be linked to the Pumped Flow slot of a Pumped Storage object.

☛ Inflow

Type: Multi Slot
Units: FLOW
Description: inflow into the reservoir from upstream
Information:
I/O: Optional; if not input by the user, it is set through either mass balance computations or the propagation of values across the link.
Links: May be linked to one or more outflow slots of upstream objects.

☛ Inflow 2

Type: Series Slot
Units: FLOW
Description: additional inflow slot for upstream inflows that do not contribute to the wedge storage
Information:
I/O: Optional; defaults to zero if not input and not linked
Links: Can be linked to the Outflow of an upstream object.

☛ Max Iterations

Type: Table
Units: NOUNITS
Description: maximum number of allowable iterations for iterative loops in the solution algorithms
Information: Used in conjunction with Convergence Percentage as a stopping criterion for iterative calculations.
I/O: Optional; defaults to 100 if not input.

Links: Not linkable

☛ Operating Head

Type: Series Slot
Units: LENGTH
Description: elevation difference between the average Pool Elevation and the average Tailwater Elevation during a timestep
Information:
I/O: Output only
Links: Usually not linked

☛ Outflow

Type: Series Slot
Units: FLOW
Description: outflow from reservoir
Information: The outflow from a is equal to the sum of the Turbine Release and the Spill. May be linked to the inflow slot of a downstream object. If not input by the user, it is set through either the mass balance computations or the propagation of values across the link.
I/O: Optional; if not input by the user, it is set through either the mass balance computations or the propagation of values across the link.
Links: May be linked to the inflow slot of a downstream object.

☛ Pool Elevation

Type: Series Slot
Units: LENGTH
Description: elevation of the water surface of the Reservoir
Information: There must be an initial value for either Storage or Pool Elevation given by the user for the first timestep.
I/O: Optional; if not input by the user, it is solved by the mass balance computations. It may take a TARGET flag indicated by the user for target operation solution.
Links: May be linked to Tailwater Elevation or Tailwater Base Value of an upstream object or to Elevation 1 or Elevation 2 of a Canal object.

☛ Power

Type: Series Slot
Units: POWER
Description: power generated by flow through the turbines
Information: Calculated by the power methods and cannot be input by the user.
I/O: Output only

Links: Not linkable

☛ RETURN FLOW

Type: Multi Slot
Units: FLOW
Description: flow returning from a diverting object
Information:
I/O: Optional; defaults to zero if not linked or input.
Links: May be linked to one or more Return Flow slots on Water User objects or the Total Return Flow slot on the Agg Diversion Site objects.

☛ SPILL

Type: Series Slot
Units: FLOW
Description: sum of the regulated and unregulated spills and bypass
Information: May be input or solved for by **RiverWare**™ (see spill calculation methods).
I/O: Optional; may be input or solved for by **RiverWare**™ (see spill calculation methods).
Links: Usually not linked

☛ STORAGE

Type: Series Slot
Units: VOLUME
Description: volume of water stored in the reservoir
Information: May be flagged as a TARGET Storage value by the user. There must be an initial value for either Storage or Pool Elevation given by the user for the first timestep. If flagged as a TARGET, a target operation solution is used.
I/O: Optional; if not input by the user, it is set through mass balance computations.
Links: Usually not linked

☛ TAILWATER ELEVATION

Type: SeriesSlot
Units: LENGTH
Description: water surface elevation on the downstream side of the dam
Information: It can be linked to the Pool Elevation or Backwater Elevation of a downstream reservoir if the “Linked or Input” method is selected for the Tailwater category. Otherwise, it is calculated by the user method selected. It is used to compute the Operating Head used in the power calculations.
I/O: Optional; can be input, linked or calculated.

Links: It can be linked to the Pool Elevation or Backwater Elevation of a downstream reservoir if the "“Linked or Input” method is selected for the Tailwater category.

☛ TOTAL INFLOWS

Type: SeriesSlot
Units: FLOW
Description: Summary slot displaying the flows into and out of the reservoir excluding the flows through the outlet works
Information: Total Inflows is calculated using the following equation:

Total Inflows = Inflow + Canal Flow +
Hydrologic Inflow + Hydrologic Inflow Adjust + Hydrologic Inflow Forecast +
Return Flow + Flow FROM Pumped Storage – Flow TO Pumped Storage – Diversion

Any component that is not in use or is not valid defaults to zero.
I/O: Output only
Links: Not linkable

☛ INFLOW SUM

Type: Series
Units: FLOW
Description: Sum of the total flows entering the reservoir at each timestep
Information: Inflow Sum is calculated using the following equation:

Inflow Sum = Inflow + Canal Flow +
Hydrologic Inflow + Hydrologic Inflow Adjust + Hydrologic Inflow Forecast +
Return Flow + Flow FROM Pumped Storage

I/O: Output only
Links: Not Linkable

☛ TURBINE RELEASE

Type: Series Slot
Units: FLOW
Description: flow through the turbines of a power reservoir (excluding spill)
Information:
I/O: Optional; solved for if not input.
Links: Usually not linked

Note: The initial (Beginning of Run) value for either Pool Elevation or Storage must be input by the user.

23.1 User Methods

23.1.1 Power

The Power category user methods calculate the flow through the turbines (Turbine Release) and the Power and Energy generated. These methods require that the total Outflow of the Reservoir be known

23.1.1.1 None

This is the default method in the Power category. It contains no calculations for Power or Energy. There are no slots specifically associated with this method.

SLOTS SPECIFIC TO THIS METHOD

☛ **NONE**

The method first checks that Energy and Turbine Release are not input by the user. These slots cannot be input when None is the selected method. If either of these two slots are input, a **RiverWare**™ error will be posted and the simulation run aborted.

Next, the selected method in the Power Plant Failure category is executed. This sets the Power Plant Cap Fraction if necessary and checks for plant shutoff/failure. If the plant is shutoff/failed, the turbine Release is set to 0.0.

Otherwise, the Turbine Release is calculated as the difference between Outflow and Spill.

23.1.1.2 No Power Turbine Flow

The No Power Turbine Flow method is used to model Turbine Release without any power generation. Turbine Release is calculated as the Outflow minus Spill. The computed Turbine Release can not be larger than the Max Flow Through Turbines.

SLOTS SPECIFIC TO THIS METHOD

☛ **MAX FLOW THROUGH TURBINES**

Type: Table Slot
Units: LENGTH VS. FLOW
Description: relationship between Pool Elevation and Turbine Capacity
Information:
I/O: Required input
Links: Not linkable

The method first checks that Energy and Turbine Release are not input by the user. These slots are not valid for user input when the No Power Turbine Flow method is selected. If either of these two slots is input, a **RiverWare**™ error will be posted and the simulation run will be aborted. Pool Elevation is then used in an interpolation scheme to determine the maximum release from the Max Turbine Flow table.

Next, the selected method in the Power Plant Failure category is executed. This sets the Power Plant Cap Fraction if necessary and checks for plant shutoff/failure. If the plant is shutoff/failed, the turbine Release is set to 0.0.

Otherwise, the Turbine Release is set as either Outflow minus Spill or maximum release. It is set as the lesser of the two values because the Turbine Release must be less than the Turbine Capacity.

23.1.1.3 Plant Power Coefficient

The Plant Power Coefficient method calculates the Power and Energy generated based on the whole plant characteristics. If the Power Coefficient is specified, the Power is calculated directly, unless the **BEST EFFICIENCY** or **MAX CAPACITY** flag is set on Energy. If its not input, the Power Coefficient is found from the interpolation of the Best or Max Turbine Q and Power Coefficient tables using the current Operating Head. If the Turbine Release is less than the Best Turbine Q, the Best Power Coefficient Table is used. If the Turbine Release is greater than the Max Turbine Q, then the Max Power Coefficient Table is used. If the Turbine Release is between the two, an intermediate Power Coefficient Value is found by interpolation.

Slots Specific to this Method

☛ Best Hydro Capacity

Type: Series Slot
Units: POWER
Description: most efficient hydro capacity of the plant at the current timestep
Information: Solved iteratively based on Best Turbine Q and the Best Power Coefficient.
I/O: Output only
Links: Not linkable

☛ Best Power Coefficient

Type: Table Slot
Units: LENGTH VS. POWER PER FLOW
Description: Operating Head vs. most efficient power coefficient
Information: The Power Coefficient relates turbine release to power generated. The Best Power Coefficient represents the most efficient power generation.
I/O: Required input

Links: Not linkable

☛ Best Turbine Q

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Operating Head vs. flow through turbine for most efficient power generation
Information:
I/O: Required input
Links: Not linkable

☛ Hydro Capacity

Type: Agg Series Slot
Units: POWER
Description: maximum hydro capacity of plant at the current timestep
Information: Solved iteratively based on Max Turbine Q and the Maximum Power Coefficient.
I/O: Output only
Links: Usually not linked

☛ Max Power Coefficient

Type: Table Slot
Units: LENGTH VS. POWER PER FLOW
Description: Operating Head vs. maximum power coefficient
Information: The Power Coefficient relates turbine release to power generated. The Max Power Coefficient represents the maximum Turbine Release.
I/O: Required input
Links: Not linkable

☛ Max Turbine Q

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Operating Head vs. maximum flow through the turbine
Information:
I/O: Required input
Links: Not linkable

☛ Minimum Power Elevation

Type: Table Slot
Units: LENGTH
Description: minimum Pool Elevation for power production

Information:
I/O: Required input
Links: Not linkable

☛ Plant Power Limit

Type: Series Slot
Units: POWER
Description: Power output is limited to this value
Information:
I/O: Optional; This constraint on power is only applied if the user inputs a value for the timestep.
Links: Not linkable

☛ Power Coefficient

Type: Series Slot
Units: POWER PER FLOW
Description: power generated per unit power release
Information:
I/O: Optional; can be input or calculated.
Links: Usually not linked

☛ Power Plant Cap Fraction

Type: Series Slot
Units: FRACTION
Description: the percentage of full capacity of the turbine units in the hydropower plant
Information:
I/O: Optional; The value of this slot defaults to 100% if not input by user.
Links: Not linkable

This method performs calculations to compute the power generated at each timestep.

First, Tailwater Elevation and Operating Head are determined based on the user method selected in the Tailwater category.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the Minimum Power Elevation was input by the user. If no value was input, a **RiverWare**™ error is posted and the simulation run is aborted. If the previous Pool Elevation is less than the Minimum Power Elevation or the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

4. Energy, Power, Hydro Capacity, and Best Hydro Capacity are set equal to zero.

5. If the Turbine Release is input or already set from the Dispatch Method “solveMB_givenInflowRelease,” a **RiverWare**™ error is flagged and the run is aborted.

6. Turbine Release is set equal to zero.

Now, Operating Head is used to determine the maximum power release through interpolation on the Max Turbine Q table. The maximum power release value is multiplied by the Power Plant Cap Fraction to account for the state of the turbine units.

If Turbine Release is already set from the dispatch method “solveMB_givenInflowRelease”, the following checks are performed:

- If Turbine Release is greater than Outflow - Spill, a **RiverWare**™ error is posted reading, “Requested Power Release is Greater than Outflow - Spill” and the run is aborted.

- If Turbine Release is greater than the maximum power release, a **RiverWare**™ error is posted reading, “Requested Turbine Release is greater than Maximum Turbine Capacity” and the run is aborted.

If the Turbine Release was input by the user, a **RiverWare**™ error is posted and the run is aborted. If neither the Energy nor the Turbine Release were input and the Energy was not set by a rules, the Turbine Release is set equal to the lesser of the Maximum Power Release or the value of the Outflow minus the Spill.

Using the calculated value of Operating Head, QmaxTemp and QbestTemp are obtained from the Max Turbine Q Table and the Best Turbine Q Table, respectively. Both values are then multiplied by the Power Plant Cap Fraction to obtain Qmax and Qbest. The Operating Head is also used to determine both the best power coefficient and the max power coefficient through interpolation of the Best Power Coefficient and Max Power Coefficient tables, respectively.

The following calculations are not performed if Energy is Input, set by a Rule, or flagged **BEST EFFICIENCY** or **MAX CAPACITY**. In these cases the Power, Energy, and Power Coefficient have already been calculated in Plant Power Coefficient Release.

If the Power Coefficient is not input by the user, the following steps are performed:

1. If the maximum power coefficient is greater than the best power coefficient, the following **RiverWare**™ error is posted, “best Power Coeff < full gate Power Coeff” and the simulation run is aborted.

2. If Qbest is greater than Qmax, the following **RiverWare**™ error is posted, “Best Turbine Q > Max Turbine Q” and the simulation run is aborted.

3. If Qbest equals Qmax, the Power Coefficient is set equal to the best power coefficient.

4. If none of the previous three conditions are satisfied and the Turbine Release is less than or equal to Qbest, the Power Coefficient is set equal to the best power coefficient.

5. If none of the previous four conditions are satisfied and the Turbine Release is less than Qmax, the Power Coefficient is calculated using the following equation:

$$\text{Power Coefficient} = \text{best power coefficient} + \frac{(\text{Turbine Release} - \text{Qbest})}{(\text{Qmax} - \text{Qbest})} \times (\text{max power coefficient} - \text{best power coefficient})$$

6. If none of the previous four conditions are true, the Power Coefficient is set equal to the max power coefficient.

Power is then calculated using the following equation:

$$\text{Power} = \text{Power Coefficient} \times \text{Turbine Release}$$

If the user has input the Plant Power Limit, the following steps are taken:

1. If the Power Coefficient is input by the user, Power and Turbine Release may need to be recalculated. If the Power is greater than the Plant Power Limit, Power is set equal to the Plant Power Limit and Turbine Release is recalculated as the Plant Power Limit divided by the Power Coefficient.

2. If the Power Coefficient is not input and the Plant Power Limit is exceeded; the Turbine Release, Power, and Power Coefficient may need to be recalculated. If the Power Coefficient is equal to the best power coefficient, the plant is already operating at best efficiency. Therefore, the Turbine Release is set equal to the Plant Power Limit divided by the Power Coefficient and the rest of the flow is spilled. The Power and Power Coefficient do not need to be recalculated.

If the Power Coefficient is not equal to the best power coefficient, Turbine Release, Power, and the Power Coefficient need to be recalculated. This is done through the following steps:

- Temporary variables are calculated from the following equations:

$$\text{power at best} = \text{best power coefficient} \times \text{Qbest}$$

$$\text{power at max} = \text{max power coefficient} \times \text{Qmax}$$

$$\text{Qlimit} = \text{Qbest} + \frac{\text{Plant Power Limit} - \text{power at best}}{(\text{power at max} - \text{power at best}) \times (\text{Qmax} - \text{Qbest})}$$

$$\text{PClimit} = \text{best power coefficient} + \frac{\text{Plant Power Limit} - \text{power at best}}{(\text{power at max} - \text{power at best}) \times (\text{max power coefficient} - \text{best power coefficient})}$$

- If Qlimit is greater than Qmax: the Power Coefficient is set equal to the max power coefficient, Turbine release is set equal to Qmax, and Power is set equal to the power at max.
- If Qlimit is less than Qbest: the Power Coefficient is set equal to the best power coefficient, Turbine Release is set equal to Qlimit, and Power is set equal to the Plant Power Limit.
- If Qlimit is less than Qmax but greater than Qbest: the Power Coefficient is set to equal to PClimit, Turbine Release is set equal to Qlimit, and Power is set equal to the Plant Power Limit.

If the Spilled Energy Power Coefficient is visible on the reservoir and it is not input, it is set equal to the Power Coefficient.

Energy is calculated as Power multiplied by the timestep (in hours).

The following calculations take always take place, regardless of the flag on Energy:

If either the Turbine Release is equal to the maximum power release or the Energy is at the maximum capacity, Hydro Capacity is set equal to Power and the Best Hydro Capacity is obtained from the getHydroCap function. If the energy is at the **Best Efficiency**, the Best Hydro Capacity is set equal to the Power and the Hydro Capacity is obtained from the getHydroCapacity function. If neither the Turbine Release is equal to the maximum power release, the Energy is at the maximum capacity, nor the Energy is at the **Best Efficiency**, both Hydro Capacity and Best Hydro capacity are obtained with the getHydroCap function.

23.1.1.4 Plant Efficiency Curve

The Plant Efficiency Curve method calculates the Power and Energy generated based on the whole plant characteristics. If the Power Coefficient is specified, the Power is calculated directly, unless the **BEST EFFICIENCY** or **MAX CAPACITY** flag is set on Energy. If the Power Coefficient is not input, the Power is found by a 3-D interpolation of the Plant Power Table using the current, average Operating Head and Turbine Release. The Power Coefficient is calculated as Power divided by Turbine Release.

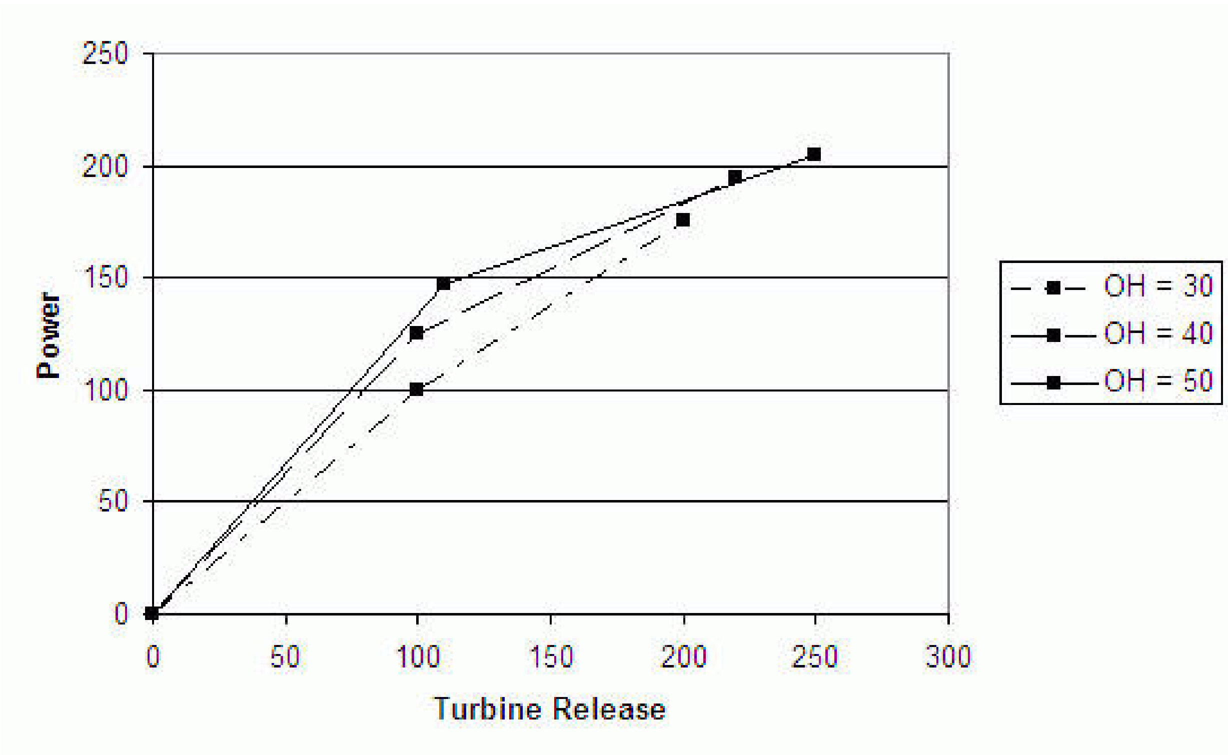
SLOTS SPECIFIC TO THIS METHOD

☛ PLANT POWER TABLE

Type: Table Slot
Units: LENGTH, FLOW, POWER
Description: 3-D table used to determine power using interpolation
Information: Data must be entered into the table in increasing, concave blocks of the same Operating Head for the 3-dimensional table interpolation to work correctly. For every block of the same Operating Head in column 1, Turbine Release should be listed in increasing, concave order in column 2, and the

corresponding Power in column 3. There should also be a point of zero Turbine Release and zero Power for each operating head. The second to last row for each operating head is the point of best efficiency. The last row for each operating head is the point of maximum Turbine Release and maximum Power production. If there are only two rows for a given operating head, both the **best efficiency** and **max capacity** are equal to the second row. The table shown below is an example of the proper way to formulate the Plant Power Table. The graph displays the increasing concave blocks at each operating head.

Operating Head	Turbine Release	Power
30	0	0
30	100	100
30	200	175
40	0	0
40	100	125
40	220	195
50	0	0
50	110	147
50	250	205



I/O: Input Only

Links: Not Linkable

☛ Power Coefficient

Type: Series Slot
Units: POWER PER FLOW
Description: power generated per unit flow release
I/O: Optional; if input, it is used to compute power. Otherwise, power is computed from the Plant Power Table

Links: Not usually linked

☛ Hydro Capacity

Type: Agg Series Slot
Units: POWER
Description: This is the maximum power that can be produced at the current timestep
Information: Solved for iteratively based on the Operating Head and maximum possible release.

I/O: Output Only

Links: Not Linkable

☛ Best Hydro Capacity

Type: Series Slot
Units: POWER
Description: This is the power that would be produced at the most efficient operating point at the current timestep.
Information: Solved for iteratively based on the most efficient operating point and the corresponding release.
I/O: Output Only
Links: Not Linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: FRACTION
Description: Must be a number less than or equal to 1. If not input, automatically set to 1.
Information: This is the percentage of full capacity of the turbine units in the hydropower plant. For example, if only half of the turbines are operational (and they are all the same), this value would be 0.5
I/O: Can be input by user. If not, value is set to 1.
Links: Not Linkable

☛ Minimum Power Elevation

Type: Series Slot
Units: LENGTH
Description: The minimum elevation at which the reservoir can still produce power.
I/O: Input Only
Links: Not Linkable

☛ Plant Power Limit

Type: Series Slot
Units: POWER
Description: The max power that the plant can produce at a given timestep.
I/O: Optional, only applies if input by user
Links: Not Linkable

☛ Power Curvature Tolerance

Type: Scalar
Units: NONE

Description: The power curvature tolerance is used to account for anomalies in Plant Power Table data and round off error while calculating slopes.
Information: Although the units for the slot are "None", the comparison is implicitly using (MW/cms).
I/O: Input or defaults to $1X10^{-6}$
Links: Not linkable

At the start of an optimization run, the Plant Power Table is checked for concavity. The slope of each segment for each block is calculated as:

$$Slope_{Segment} = \frac{Power_i - Power_{i-1}}{TurbineRelease_i - TurbineRelease_{i-1}}$$

The table is considered concave if:

$$Slope_{Segment} \leq Slope_{PreviousSegment} + PowerCurvatureTolerance$$

The table is required to be concave for optimization runs, but not for simulation or rulebased simulation runs.

This method performs calculations to compute the power generated at each timestep.

First, Tailwater Elevation and Operating Head are determined based on the user method selected in the Tailwater category.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the Minimum Power Elevation was input by the user. If no value was input, a **RiverWare**™ error is posted and the simulation run is aborted. If the previous Pool Elevation is less than the Minimum Power Elevation or the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Energy, Power, Hydro Capacity, and Best Hydro Capacity are set equal to zero.
2. If the Turbine Release is input or already set from the Dispatch Method "solveMB_givenInflowRelease," a **RiverWare**™ error is flagged and the run is aborted.
3. Turbine Release is set equal to zero.

Operating Head is used to determine the maximum power release through interpolation on the Plant Power Table. The maximum power release value is multiplied by the Power Plant Cap Fraction to account for the state of the turbine units.

If Turbine Release is already set from the dispatch method "solveMB_givenInflowRelease", the following checks are performed:

- If Turbine Release is greater than Outflow - Spill, a **RiverWare**™ error is posted reading, "Requested Power Release is Greater than Outflow - Spill" and the run is aborted.
- If Turbine Release is greater than the maximum power release, a **RiverWare**™ error is posted reading, "Requested Turbine Release is greater than Maximum Turbine Capacity" and the run is aborted.

If the Turbine Release was input by the user, a **RiverWare**™ error is posted and the run is aborted. If neither the Energy nor the Turbine Release were input and the Energy was not set by a rules, the Turbine Release is set equal to the lesser of the Maximum Power Release or the value of the Outflow minus the Spill.

The following calculations are not performed if Energy is Input, set by a Rule, or flagged **BEST EFFICIENCY** or **MAX CAPACITY**. In these cases the Power and Energy have already been calculated in Plant Efficiency Curve Release.

If the Power Coefficient is input by the user,

$$Power = TurbineRelease \times PowerCoefficient$$

Otherwise, Power is found directly from the Plant Power Table using the current Operating Head and the Turbine Release from above. The power coefficient is now calculated as:

$$PowerCoefficient = Power / TurbineRelease$$

If the user has input the Plant Power Limit, the following steps are taken:

1. If the Power Coefficient is input by the user, Power and Turbine Release may need to be recalculated. If the Power is greater than the Plant Power Limit, Power is set equal to the Plant Power Limit and Turbine Release is recalculated as the Plant Power Limit divided by the Power Coefficient.

2. If the Power Coefficient is not input and the Plant Power Limit is exceeded; the Turbine Release, Power, and Power Coefficient need to be recalculated. The Power is set equal to the Plant Power Limit and the Turbine Release is found using 3-D interpolation of the Plant Power Table. The Power Coefficient is then calculated as Power divided by Turbine Release.

Energy is then calculated as Power multiplied by the timestep length.

The following calculations take always take place, regardless of the flag on Energy:

If either the Turbine Release is equal to the maximum power release or the Energy is at the maximum capacity, Hydro Capacity is set equal to Power and the Best Hydro Capacity is computed iteratively. If the energy is at the **Best Efficiency**, the Best Hydro Capacity is set equal to the Power and the Hydro Capacity is computed iteratively. If the Turbine Release is not equal to the maximum power release, the Energy is not at the maximum capacity, and the

Energy is not at the **Best Efficiency**, both Hydro Capacity and Best Hydro capacity are computed by an iterative algorithm.

Notes on Power Plant Cap Fraction

If the Power Plant Cap Fraction is input by the user, it is necessary for the Plant Power Table to basically be scaled back to account for the operating points when the turbines are operating at less than 100%. To do this, when Turbine Release is known and Power is to be found using the Plant Power Curve, Turbine Release is divided by the Power Plant Cap Fraction. This point is then found in the Plant Power Curve for the current operating head and the Power is found using 3-D interpolation. Finally the Power is multiplied by the Power Plant Cap Fraction to get the actual Power produced for the current timestep.

If Power is known, and Turbine release is to be found in the table. Power is multiplied by the Power Plant Cap Fraction and then this point is found in the Plant Power Curve to solve for Turbine Release. Turbine Release is then divided by the Power Plant Cap Fraction to get the actual Turbine Release for the current timestep.

23.1.1.5 Plant Power Equation

The Plant Power Equation method is used to calculate Power and Energy using the water power equation.

Slots Specific to this Method

☛ Head Loss

Type: Table Slot
Units: LENGTH
Description: The head loss water incurs before it reaches the turbines.
Information: The slot is set to zero if not input by the user.
I/O: optional
Links: Not linkable

☛ Minimum Power Elevation

Type: Table Slot
Units: LENGTH
Description: Minimum pool elevation at which power can be generated
Information: Single value in a 1x1 table slot
I/O: Required input
Links: Not linkable

☛ Net Head vs Max Turbine Release

Type: Table Slot

Units: LENGTH VS. FLOW
Description: relationship between the net head and the maximum possible turbine release
Information: Net Head must account for any head loss
I/O: Required input
Links: Not linkable

☛ Net Head vs Plant Efficiency
Type: Table Slot
Units: LENGTH VS NONE
Description: This table allows you to specify the efficiency as a function of (previous) Net Head.
Information: This table is used only when the Plant Efficiency Value is empty. The Net Head used in this table look up comes from the previous timestep's operating head.
I/O: Optional Input
Links: Not Linkable

☛ Plant Efficiency Value
Type: Table Slot
Units: NONE
Description: the decimal percent efficiency at which the plant is operating
Information: Single value in a 1x1 table slot. Plant efficiency should incorporate both generator efficiency and turbine efficiency.
I/O: Optional Input, if specified, it must be between 0 and 1.
Links: Not linkable

☛ Power Plant Cap Fraction
Type: Series Slot
Units: NONE
Description: decimal fraction of the power capacity at which the plant is operating
Information: Used in the case of outages or reductions in the plant operating capacity.
I/O: Defaults to 1.0 if not input.
Links: Not linkable

☛ Plant Power Limit
Type: Series Slot
Units: POWER
Description: The user specified upper limit on power production
Information: If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit and the Energy is recalculated. A new Turbine Release are then calculated based on the Plant Power Limit.

I/O: Optional Input or set by a rule.

The method first checks whether Energy is either user input or set by rules. If it is, the method finishes successfully and exits-- all power calculations were already performed in the Plant Power Equation Release method.

Otherwise, Tailwater Elevation and Operating Head are determined based on the user method selected in the Tailwater category.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the Minimum Power Elevation was input by the user. If no value was input, a **RiverWare**™ error is posted and the simulation run is aborted. If the previous Pool Elevation is less than the Minimum Power Elevation or the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Energy and Power are set equal to zero.
2. If the Turbine Release is input or already set from the Dispatch Method “solveMB_givenInflowRelease,” a **RiverWare**™ error is flagged and the run is aborted.
3. Turbine Release is set equal to zero.

Once the initial checks are performed, the method calculates Net Head and Turbine Release as:

$$NetHead = OperatingHead - HeadLoss$$

Next, the Max Turbine Release for the current Net Head is interpolated from the Net Head vs. Max Turb Release table.

If there is a valid value in the Plant Efficiency Value slot, that is used for the *efficiency*. Otherwise, given the net head from the previous timestep (Operating Head at previous timestep minus Head Loss), the *efficiency* is interpolated from the Net Head vs Efficiency table. The previous Operating Head is used as an approximation so as not to introduce an additional variable in the iteration. As a result, the Tailwater Elevation at the initial timestep must be input. The net head for the initial timestep is the initial Pool Elevation minus the initial Tailwater Elevation minus Head Loss.

The method checks whether Turbine Release is user input or set by a link or a rule. If Turbine Release is known at the dispatch level, the method will check that it is not greater than Outflow minus Spill or the Max Turbine Release given the current Net Head. If either of these are true, an error will be posted and the run will abort. Otherwise, the known Turbine Release value will be used in the Power calculations. If Turbine Release is **not** user input or solved for in the dispatch methods, it is calculated as the minimum of the Max

Turbine Release (given the current Net Head) and Outflow minus Spill (either unregulated spill or user specified regulated spill):

$$TurbineRelease = Min(MaxTurbineRelease \times PowerPlantCapFraction, Outflow - Spill)$$

Note: If Turbine Release is set to MaxTurbineRelease, it means there is still some remaining water that must be passed via regulated spill. This will be calculated in the spill calculations. If Turbine Release is set to Outflow minus Spill, it means that Spill consists of Unregulated Spill and any input Regulated Spill-- all other water will pass through the turbines.

Once efficiency, Net Head, and Turbine Release are all known, Power is solved for using the Power Equation:

$$Power = \frac{\text{Turbine Release} \times \text{Net Head} \times \text{efficiency}}{\text{Unit Compatibility Factor}}$$

The unit compatibility factor comes from balancing units and is 102.01697767 in internal RiverWare units.

If the computed Power is greater than the Plant Power Limit, the Power is reset to the Plan Power Limit and Turbine Release is recomputed by solving the above equation for Turbine Release.

Lastly, Energy is computed as Power multiplied by the time length of the timestep.

$$Energy = \text{Power} \times \text{Length of Timestep}$$

23.1.1.6 Unit Generator Power

The Unit Generator Power method is used to calculate Power and Energy for generating units with individual characteristics. The generating units are grouped by unit type for ease of data entry.

SLOTS SPECIFIC TO THIS METHOD

☛ BEST GENERATOR FLOW

Type: Table Slot
Units: LENGTH VS. FLOW
Description: a table for each unit type which gives the relationship between operating head and flow through the generator when operating at best efficiency
Information: There must be a block of data for each unit type given in the Generator Unit Types table. The table is representative of a single unit within the specified unit type.
I/O: Required input
Links: Not linkable

☛ Best Generator Power

Type: Table Slot
Units: LENGTH VS. POWER
Description: a table for each unit type which gives the relationship between operating head and power produced by the generator when operating at best efficiency
Information: There must be a block of data for each unit type given in the Generator Unit Types table. The table is representative of a single unit within the specified unit type.
I/O: Required input
Links: Not linkable

☛ Full Generator Flow

Type: Table Slot
Units: LENGTH VS. FLOW
Description: a table for each unit type which gives the relationship between operating head and flow through the generator when operating at full capacity
Information: There must be a block of data for each unit type given in the Generator Unit Types table. The table is representative of a single unit within the specified unit type.
I/O: Required input
Links: Not linkable

☛ Full Generator Power

Type: Table Slot
Units: LENGTH VS. POWER
Description: a table for each unit type which gives the relationship between operating head and power produced by the generator when operating at full capacity
Information: There must be a block of data for each unit type given in the Generator Unit Types table. The table is representative of a single unit within the specified unit type.
I/O: Required input
Links: Not linkable

☛ Generator Unit Types

Type: Table Slot
Units: NONE
Description: a list of each generating unit and the corresponding unit type
Information: More than one generating unit can be assigned to a given unit type. The unit type must be an integer value beginning with 1 and increasing by increments of 1.
I/O: Required input

Links: Not linkable

☛ Generators Available and Limit

Type: Table Series Slot
Units: FRACTION AND POWER
Description: a time series specifying the availability and power limit of each generating unit.
Information: Availability is a number between 0 and 1 which represents the percentage of the timestep that the unit is available. There must be a block of data for each row in the Generator Unit Types table. The Power Limit has no effect on the flow through the turbines.
I/O: Required input
Links: Not linkable

☛ Hydro Capacity

Type: Agg Series Slot
Units: POWER
Description: the maximum power production possible at the current timestep
Information: This value is the sum of all generators operating at full capacity for the given operating head at the current timestep.
I/O: Output only
Links: Could be linked to a Data Object, but usually not linked.

☛ Minimum Power Elevation

Type: Table Slot
Units: LENGTH
Description: the minimum pool elevation required for power production
Information: When the Pool Elevation drops below this value, a warning is posted and no power is produced.
I/O: Required input
Links: Not linkable

☛ Power Coefficient

Type: Series Slot
Units: POWER/FLOW
Description: power generated per unit power release
Information: This coefficient corresponds to the efficiency of the entire plant. It is not used in calculation and is displayed only for the benefit of the user.
I/O: Output only
Links: Could be linked to a Data Object, but usually not linked.

The Unit Generator Power method begins by computing the availability and power limits of each unit type. Availability and power limit values are computed as the sum of the values from the availability and power limit columns, respectively, in the Generators Available and Limit slot. A value for availability and power limit is computed for each unit type.

First, Tailwater Elevation and Operating Head are determined based on the user method selected in the Tailwater category.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction if necessary (the default is 1.0) and checks for plant shutoff/failure.

If the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Energy and Power are set equal to zero.
2. If the Turbine Release is input or already set from the Dispatch Method "solveMB_givenInflowRelease," a **RiverWare**™ error is flagged and the run is aborted.
3. Turbine Release is set equal to zero.

Then the efficiency of each unit type is calculated by the following equation:

$$efficiency = \frac{powerTemp}{flowTemp}$$

PowerTemp and flowTemp, both local variables, are computed from the Best Generator Power and Best Generator Flow tables, respectively, using the current Operating Head. Each unit type is then sorted in descending order based on the computed efficiency.

Turbine Release has already been computed by the dispatch method and the power produced from the known Turbine Release must be calculated. The method begins to add entire unit types (operating according to the best flow and power tables and beginning with the most efficient type) until the Turbine Release is exceed or all the unit types have been added. If the power generated by a particular unit type exceeds the power limit for that unit type, the power produced from that type is set to the power limit. The power limit has no effect on the flow going through the turbines. If the Turbine Release is exceeded, the last generator type is interpolated to compute the Power exactly (see equation below). However, if all the unit types have been added and the Turbine Release cannot be met, the method assumes all unit types are operating at full capacity (according to the Full Generator Flow and Full Generator Power tables). Then if the Turbine Release is exceeded, the last generator type added is interpolated to compute the Power exactly (see equation below). However, if the Turbine Release still cannot be met, all unit types are run at full capacity. Turbine Release is reset to the maximum flow through the turbines and Power is set as the maximum power produced by the turbines (at the given operating head). The spill must be recalculated to handle the excess Turbine Release that could not be met.

The interpolation equation used to calculate Power is given below:

$$\text{Power} = \text{oneLessTypePower} + \frac{\text{Turbine Release} - \text{oneLessTypeFlow}}{\text{cumulativeFlow} - \text{oneLessTypeFlow}} \bullet (\text{cumulativePower} - \text{oneLessTypePower})$$

where oneLessTypePower is the power produced from all the previous types added (excluding the most recent type added); oneLessTypeFlow is the flow through all the previous unit types (excluding the most recent type added); cumulativePower is the power produced from all the unit types added (including the most recent type); and cumulativeFlow is the flow through all the unit types added (including the most recent type).

Note: The above equation assumes the relationship between power and flow is linear, regardless of the actual relationship specified in the power and flow tables. It is also interpolating over an entire type of generators.

The Power Coefficient is then calculated by the following equation:

$$\text{Power Coefficient} = \frac{\text{Power}}{\text{Turbine Release}}$$

If all the unit types were added, the Spilled Energy Power Coefficient is equal to the Power Coefficient. However, if all the types were not added, the Spilled Energy Power Coefficient is set equal to the efficiency of the last type added.

Energy is calculated as the product of the Power and the timestep length. Hydro Capacity is set as the power produced from all units operating at full capacity.

23.1.1.7 Peak and Base

The Peak and Base method computes the Power and Energy generated by the entire plant based on the fraction of each timestep operated at peak flow and base flow. It is a long timestep method, modeled after the U. S. Bureau of Reclamation’s CRSS peak-base power calculation. A peaking flow value is first determined from the Outflow, Tailwater Elevation and Best Generator Flow. A minimum Base Flow and Power production are assumed for the entire timestep. Next, the number of hours to operate at peak power is calculated from the remaining volume of water released during that timestep. Peak production and base production are then added to determine the total Energy. Power is calculated by dividing the Energy by the timestep length in hours. Power Capacity is the power that could be generated if the flow is directed through the turbine(s) given an operating head. This is added to distinguish between actual power production and the power that could be produced.

SLOTS SPECIFIC TO THIS METHOD

☛ **BASE FLOW**

Type: Series Slot

Units: FLOW
Description: minimum flow through turbines to produce energy
Information: This value is read from the Base Flow Table
I/O: Output only
Links: Not linkable

☛ Base Flow Table

Type: Table Slot
Units: FLOW vs. FLOW
Description: Outflow from the Reservoir vs. base flow
Information: This table gives the minimum flow required through the turbines as a function of the average total outflow from the Reservoir.
I/O: Required input
Links: Not linkable

☛ Best Generator Flow

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Operating Head vs. flow through the turbine at best efficiency
Information: The minimum and maximum values of operating head in this table are used as limiting values. The operating head is reset to the min. or max if it exceeds these constraints.
I/O: Required input
Links: Not linkable

☛ Best Generator Power

Type: Table Slot
Units: LENGTH vs. POWER
Description: Operating Head vs. power at best efficiency
Information: power produced by the entire plant at base energy flow
I/O: Required input
Links: Not linkable

☛ Maximum Turbine Power

Type: Table Slot
Units: POWER
Description: maximum turbine power output
Information:
I/O: Required input
Links: Not linkable

☛ Min and Max Operating Head

Type: Table Slot
Units: LENGTH
Description: the minimum and maximum operating head for the turbines
Information:
I/O: Required input
Links: Not linkable

☛ Number of Units

Type: Table Slot
Units: NONE
Description: integer number of turbines in plant
Information:
I/O: Required input
Links: Not linkable

☛ Off Peak Capacity

Type: Table Slot
Units: POWER
Description:
Information:
I/O: Required input
Links: Not linkable

☛ Peak Flow

Type: Series Slot
Units: FLOW
Description: most efficient flow through turbines for the current Operating Head
Information:
I/O: Output only
Links: Not linkable

☛ Peak Hours

Type: Series Slot
Units: TIME
Description: the number of hours operated at peak flow
Information:
I/O: Output only
Links: Not linkable

☛ Plant Efficiency

Type: Series Slot
Units: FRACTION
Description: ratio of actual power produced to peak and base theoretical power
Information:
I/O: Output only
Links: Not linkable

☛ Power Capacity

Type: Agg Series Slot
Units: POWER
Description: power that could be produced if flow is directed through the turbines given the operating head.
Information: Calculated by the two peak power methods and cannot be input by the user.
I/O: Output only
Links: Not linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: FRACTION
Description: the percentage of full capacity of the turbine units in the hydropower plant
Information:
I/O: Optional; the value of this slot defaults to 100% if not input by user
Links: Not linkable

First, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction if necessary (the default is 1.0) and checks for plant shutoff/ failure.

If the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Energy, Power, Base Flow, Peak Flow, Peak Power, Power Capacity, and Plant Efficiency are set equal to zero.

2. If the Turbine Release is input or already set from the Dispatch Method "solveMB_givenInflowRelease," a **RiverWare**™ error is flagged and the run is aborted.

3. Turbine Release is set equal to zero.

4. Tailwater and Operating Head are computed and set. No further computations are performed.

If either the Energy or Turbine Release is input by the user, a **RiverWare™** error is posted and the simulation run is aborted. These are not valid input slots for the Peak and Base method. Peak Flow and Peak Power are then calculated as follows:

1. If either the Maximum Turbine Power is not valid or it is less than 0.00000001 MW, a **RiverWare™** Error is flagged and the simulation run is aborted.

2. The Maximum Turbine Power is then used with the Best Generator Power table to obtain a value for the local variable, headAtMaxPower.

3. headAtMaxPower is used with the Best Generator Flow table to obtain a value for the local variable, flowAtMaxPower. The local variable, flow is temporarily set as flowAtMaxPower.

4. The local variable, efficiencyAtMaxPower is calculated by the following formula:

$$\text{efficiencyAtMaxPower} = \text{Maximum Turbine Power} \times \frac{1000000}{\text{flowAtMaxPower} \times \text{headAtMaxPower} \times 999.99 \times 9.79908}$$

where 999.99 is the density of water (Kg/M^3) at five degrees C and 9.79908 is gravitational acceleration (M/s^2) at 37 degrees North latitude.

5. **RiverWare™** then iterates while:

- The absolute difference between Qnew and flow is greater than 5 cfs.
- The number of iterations is less than the maximum number of iterations.
- The Operating Head is greater than the Minimum Operating Head.

The following calculations and evaluations are inside the iterative loop:

- The local variable, Qnew is set equal to flow
- The local variable, plantFlow is determined using the following equation:

$$\text{plantFlow} = \text{Qnew} \times \text{Number of Units} \times \text{Power Plant Cap Fraction}$$

- Flow is set equal to plantFlow
- The user selected Tailwater calculation is performed
- The Operating Head is calculated
- If the Operating Head is greater than the Maximum Operating Head, Operating Head is set equal to the maximum Operating Head
- If the Operating Head is less than the Minimum Operating Head, Operating Head is set equal to the Minimum Operating Head

- If the Operating Head is greater than the headAtMaxPower, the flow is calculated using the following equation:

$$\text{flow} = \text{Maximum Turbine Power} \times \frac{1000000}{\text{Operating Head} \times \text{efficiencyAtMaxPower} \times 999.99 \times 9.79908}$$

- If the Operating Head is less than the Maximum Operating Head, greater than the Minimum Operating Head, and less than the headAtMaxPower, flow is obtained from the Best Generator Flow table and the Operating Head.

This set of calculations is repeated until the iteration criteria described above are met.

6. If Operating head is less than the Minimum Operating Head; Turbine Release, Energy, Power, Power Capacity, Peak Flow, Peak Hours, and flow are all set to zero. Then the Tailwater method is re-executed. If Operating Head is greater than the minimum Operating Head and the headAtMaxPower, Peak Power is set equal to Maximum Turbine Power. If the Operating Head is greater than the Minimum Operating Head and less than the headAtMaxPower, Peak Power is determined from the Best Generator Power table using Operating Head.

Once Peak Power and Peak Flow (called “flow” in calculations described above) are calculated, Base Power and Base Flow can be determined. Base Power is set equal to Off Peak Capacity and Base Flow is determined from Outflow and the Base Flow Table.

If Outflow minus Unregulated Spill is greater than the product of Peak Flow, Number of Units, and Power Plant Cap Fraction; the following steps are taken:

1. The Tailwater method selected by the user is executed.
2. The Operating Head is calculated.
3. The local variable, headAtMaxPower, is obtained from the Best Generator Power table using the Maximum Turbine Power.
4. The local variable, flowAtMaxPower, is obtained from the Best Generator Flow table using the headAtMaxPower.
5. The local variable, efficiencyAtMaxPower, is computed using the following formula:

$$\text{efficiencyAtMaxPower} = \frac{\text{Maximum Turbine Power} \times 1000000}{(\text{flowAtMaxPower} \times \text{headAtMaxPower} \times 999.99 \times 9.79908)}$$

6. If Operating Head is greater than headAtMaxPower, Peak Flow and Peak Power are calculated using the following equations:

$$\text{Peak Flow} = \text{Maximum Turbine Power}$$
$$\times \frac{1000000}{\text{Operating Head} \times \text{EfficiencyAtMaxPower} \times 999.99 \times 9.79908}$$

$$\text{Peak Power} = \text{Maximum Turbine Power}$$

7. If the Operating Head is less than or equal to the headAtMaxPower, Peak Flow and Peak Power are determined using the Operating Head in conjunction with the Best Generator Flow and Best Generator Power tables, respectively.

The Peak Flow slot represents the flow through the entire power plant. Therefore, the value in this slot is calculated as Peak Flow times Number of Units times Power Plant Cap Fraction. In the calculations that follow, Peak Flow represents the slot value just calculated. Plant Peak Power is calculated as Peak Power times Number of Units times Power Plant Cap Fraction.

The number of hours required to operate at base and peak flows are computed next using the following equations:

$$\text{Peak Flow Volume} = (\text{Outflow} - \text{Spill} - \text{Base Flow}) \times \text{timestep (in seconds)}$$

$$\text{Peak Hours} = \frac{\text{Peak Flow Volume}}{(\text{Peak Flow} - \text{Base Flow}) \times 3600}$$

$$\text{Base Hours} = \frac{\text{Timestep (in seconds)}}{3600} - \text{Peak Hours}$$

If Peak Hours is greater than the length of the timestep; Peak Hours is set equal to the timestep, Base Hours are set to zero, Turbine Release is set to Peak Flow, and Total Controlled Release is calculated as Outflow minus Unregulated Spill. If Peak Hours is less than or equal to the length of the timestep, Peak Hours and Base Hours remain as calculated by the above formulas, Turbine Release is Outflow minus Spill, and Total Controlled Release is set equal to the Peak Flow.

The theoretical and actual energy production are computed next. The local variable, peakEnergy is calculated as Peak Hours times Plant Peak Power. The local variable, baseEnergy is calculated as Base Hours times Base Power.

The local variable, bestBaseTheor (representing the theoretical, most efficient base energy) is calculated using the following equation:

$$\text{bestBaseTheor} = \text{Base Flow} \times \text{Operating Head at Base Flow} \times \text{Base Hours} \times 0.00980229$$

The local variable bestPeakTheor (representing the theoretical most efficient peak energy) is calculated using the following equation:

$$\text{bestPeakTheor} = \text{Peak Flow} \times \text{Operating Head at Peak Flow} \times \text{Peak Hours} \times 0.00980229$$

The value, 0.00980229, is a conversion factor necessary for energy to have units of megawatt-hours.

$$\text{Energy (MWH)} = 62.4\ (\text{lb/ft3}) \times Q\ (\text{cfs}) \times \text{time (hrs)} \times \text{Head (ft)} \times 1\ \text{KW} / 737.56\ (\text{ft-lb/s}) \times 1\ \text{MW} / 1000\ \text{KW}$$

For Q in cms and Head in meters the final conversion is 0.00980229.

Finally, the following slots are set:

$$\text{Power Capacity} = \text{Plant Peak Power}$$

$$\text{Energy} = \text{peakEnergy} + \text{baseEnergy}$$

$$\text{Power} = \frac{\text{Energy}}{\text{timestep length (hours)}}$$

$$\text{Plant Efficiency} = \frac{\text{Energy}}{\text{bestPeakTheor} + \text{bestBaseTheor}}$$

If a spill method is selected which utilizes the Spilled Energy Power Coefficient and this value is not input by the user, it is set as:

$$\text{Spilled Energy Power Coefficient} = \frac{\text{Power}}{\text{Turbine Release}}$$

If Turbine Release is zero, the Spilled Energy Power Coefficient is also zero.

23.1.1.8 Peak Power

The Peak Power method is similar to the Peak and Base method except that it computes power and energy based on Peak Flow only. A peaking flow value is first determined from the Outflow, Tailwater Elevation, and Best Generator Flow. The number of hours to operate at peak power is then calculated from the volume of water released during that timestep. A distinction is made between actual power production and the power that could be produced. Power Capacity is the peak power capacity. Power is calculated by dividing the energy by the timestep length in hours. There is no Base Flow power production.

Slots Specific to this Method

☛ Best Generator Flow

Type: Table Slot
Units: LENGTH vs. FLOW
Description: operating head vs. flow through the turbine at best efficiency
Information: The minimum and maximum values of operating head in this table are used as limiting values. The operating head is reset to the min. or max if it exceeds these constraints.
I/O: Required input
Links: Not linkable

☛ Best Generator Power

Type: Table Slot
Units: LENGTH vs. POWER
Description: operating head vs. power at best efficiency
Information:
I/O: Required input
Links: Not linkable

☛ Generator Efficiency

Type: Table Slot
Units: FRACTION
Description: the efficiency of the generators in producing power
Information: This value is the fraction of the maximum theoretical power which could be obtained from an ideal turbine.
I/O: Required input
Links: Not linkable

☛ Maximum Turbine Power

Type: Table Slot
Units: POWER
Description: maximum turbine power output
Information:
I/O: Required input
Links: Not linkable

☛ Min and Max Operating Head

Type: Table Slot
Units: LENGTH
Description: the minimum and maximum operating head for the turbines
Information:
I/O: Required input
Links: Not linkable

☛ Number of Units

Type: Table Slot
Units: NONE
Description: integer number of turbines in plant
Information:
I/O: Required input
Links: Not linkable

☛ Peak Flow

Type: Series Slot
Units: FLOW
Description: most efficient flow through turbines for the current Operating Head
Information:
I/O: Output only
Links: Not linkable

☛ Peak Hours

Type: Series Slot
Units: TIME
Description: the number of hours operated at peak flow
Information:
I/O: Output only
Links: Not linkable

☛ Plant Efficiency

Type: Series Slot
Units: FRACTION
Description: ratio of actual power produced to peak and base theoretical power
Information:
I/O: Output only
Links: Not linkable

☛ Power Capacity

Type: Agg Series Slot
Units: POWER
Description: power that could be produced if flow is directed through the turbines given the operating head.
Information: Calculated by the two peak power methods and cannot be input by the user.
I/O: Output only
Links: Not linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: FRACTION
Description: the percentage of full capacity of the turbine units in the hydropower plant
Information: The value of this slot defaults to 100% if not input by user.
I/O: Required input
Links: Not linkable

First, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction if necessary (the default is 1.0) and checks for plant shutoff/failure.

If the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Energy, Power, Peak Flow, Peak Hours, Power Capacity, and Plant Efficiency are set equal to zero.
2. Turbine Release is set equal to zero.
3. Tailwater and Operating Head are computed and set. No further computations are performed.

If Energy or Turbine Release are input by the user, an error is posted. These are not valid input slots for the Peak Power method. Peak Flow and Peak Power are then calculated as follows:

1. The Maximum Turbine Power is then used in the Best Generator Power table interpolate a value for headAtMaxPower.
2. headAtMaxPower is used in the Best Generator Flow table to interpolate a value for flowAtMaxPower. Flow is temporarily set as flowAtMaxPower.
3. efficiencyAtMaxPower is calculated by the following formula:

$$\text{efficiencyAtMaxPower} = \text{Maximum Turbine Power} \times \frac{1000000}{\text{flowAtMaxPower} \times \text{headAtMaxPower} \times 999.99 \times 9.79908}$$

where 999.99 is the density of water (Kg/M^3) at five degrees C and 9.79908 is gravitational acceleration (M/s^2) at 37 degrees North latitude.

4. **RiverWare™** then iterates until all of the following conditions are met; the absolute difference between Qnew and flow is less than 5 cfs, the number of iterations is greater than the maximum number of iterations, and the Operating Head is less than the Minimum Operating Head. Initially, Qnew is set equal to flow. Plant Flow is calculated as Qnew times the Number of Units times the Plant Power Cap Fraction. Then, the Tailwater method selected by the user is performed. If the Operating Head is greater than the Maximum Operating Head, the Operating Head is set equal to the Maximum Operating Head. If the Operating Head is less than the Minimum Operating Head, Operating Head is set equal to Minimum Operating Head. If the Operating Head is greater than headAtMaxPower, flow is calculated as:

$$\text{flow} = \text{Maximum Turbine Power} \times \frac{1000000}{\text{Operating Head} \times \text{efficiencyAtMaxPower} \times 999.99 \times 9.79908}$$

Otherwise, flow is determined by interpolation using Operating Head and the Best Generator Flow table. This set of calculations is repeated until the iteration criteria described above are met.

5. If Operating head is less than the Minimum Operating Head, Turbine Release, Energy, Power, Power Capacity, Peak Flow, Peak Hours, and flow are all set to zero. Then, the Tailwater method is re-executed. If Operating Head is greater than headAtMaxPower, Peak Power is set equal to Maximum Turbine Power. Otherwise, Peak Power is determined from the Best Generator Power table using Operating Head.

Once flow and Peak Power are determined, the following computations are performed:

$$\text{Peak Flow} = \text{flow} \times \text{number of Units} \times \text{Power Plant Cap Fraction}$$

$$\text{Plant Peak Power} = \text{Peak Power} \times \text{Number of Units} \times \text{Power Plant Cap Fraction}$$

$$\text{Peak Flow Volume} = (\text{Outflow} - \text{Spill}) \times \text{Timestep (in seconds)}$$

$$\text{Peak Hours} = \frac{\text{Peak Flow Volume}}{\text{Peak Flow} \times 3600 \text{ seconds}}$$

If the value of Peak Hours is greater than the length of the timestep, Peak Hours is set to the length of the timestep, Turbine Release is equal to Peak Flow, and Total Controlled Release is equal to Outflow minus Unregulated Spill. Otherwise, Peak Hours remains unchanged, Turbine Release equals Outflow minus spill, and Total Controlled Release equals Peak Flow.

Next, the theoretical and actual energy production is calculated.

$$\text{Peak Energy} = \text{Peak Hours} \times \text{Plant Peak Power} \times \text{Generator Efficiengy}$$

$$\text{Best Peak Theoretical} = \text{Peak Flow} \times \text{Operating Head} \times \text{Peak Hours} \times 0.00980229$$

The value, 0.00980229, is a conversion factor necessary for energy to have units of megawatt-hours.

$$\text{Energy (MWH)} = 62.4\ (\text{lb/ft3}) \times Q\ (\text{cfs}) \times \text{time (hrs)} \times \text{Head (ft)} \times 1\ \text{KW} / 737.56\ (\text{ft-lb/s}) \times 1\ \text{MW} / 1000\ \text{KW}$$

For Q in cms and Head in meters the final conversion is 0.00980229.

Finally, the following slots are set:

$$\text{Power Capacity} = \text{Plant Peak Power}$$

$$\text{Energy} = \text{Peak Energy}$$

$$\text{Power} = \frac{\text{Energy}}{\text{timestep length (hours)}}$$

If Best Peak Theoretical is equal to zero, Plant Efficiency is also equal to zero. Otherwise Plant Efficiency is calculated as:

$$\text{Plant Efficiency} = \frac{\text{Energy}}{\text{Best Peak Theoretical}}$$

If a spill method is selected which utilizes the Spilled Energy Power Coefficient and this value is not input by the user, it is set as:

$$\text{Spilled Energy Power Coefficient} = \frac{\text{Power}}{\text{Turbine Release}}$$

If Turbine Release is zero, the Spilled Energy Power Coefficient is also zero.

23.1.1.9 Peak Power Equation

The Peak Power Equation method provides a standard equation method of calculating plant peaking power for a portion of the computational timestep using the water power equation.

Slots Specific to this Method

☛ Head Loss

Type: Table Slot
Units: LENGTH
Description: The head loss water incurs before it reaches the turbines.
Information: The slot is set to zero if not input by the user.
I/O: optional
Links: Not linkable

☛ Min Power Elevation

Type: Table Slot
Units: LENGTH
Description: Minimum pool elevation at which power can be generated
Information: Single value in a 1x1 table slot
I/O: Required input
Links: Not linkable

☛ Net Head vs. Peak Release

Type: Table Slot
Units: LENGTH VS. FLOW
Description: relationship between the net head and the maximum possible turbine release
Information: Net Head must account for any head loss
I/O: Required input
Links: Not linkable

☛ Peak Release

Type: Series Slot
Units: FLOW
Description: The flow through the turbines when the plant is operating at peak capacity
Information: Peak Release is solved for iteratively using net head, tailwater elevation and pool elevation
I/O: Output only
Links: Not linkable

☛ Peak Times

Type: Series Slot
Units: TIME
Description: The time at which the plant is operating at peak capacity
Information: Peak Time is calculated as the timestep flow volume divided by the Peak Release.
I/O: Output
Links: Not linkable

☛ Plant Efficiency

Type: Table Slot
Units: NONE
Description: the decimal percent efficiency at which the plant is operating
Information: Single value in a 1x1 table slot. Plant efficiency should incorporate both generator efficiency and turbine efficiency.
I/O: Defaults to 1.0 if not input. Must be between 0 and 1.
Links: Not linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: NONE
Description: decimal fraction of the power capacity at which the plant is operating
Information: Used in the case of outages or reductions in the plant operating capacity.
I/O: Defaults to 1.0 if not user input. Must be between 0 and 1.
Links: Not linkable

The Peak Power Equation method first performs a series of checks.

The selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the Minimum Power Elevation was input by the user. If no value was input, a **RiverWare**™ error is posted and the simulation run is aborted. If the previous

Pool Elevation is less than the Minimum Power Elevation or the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Power, Energy, Peak Release, and Peak Time are set equal to zero.
2. If the Turbine Release is input or already set from the Dispatch Method "solveMB_givenInflowRelease," a **RiverWare**™ error is flagged and the run is aborted.
3. Turbine Release is set equal to zero.
4. Tailwater and operating head are computed. No further calculations are performed.

The method checks whether Turbine Release (the average flow through the turbines over the whole timestep) is user input or set by rules. (i.e. the method checks if the dispatch type is solveMB_givenInflowRelease.) If the given Turbine Release value is greater than Outflow minus Spill, an error is posted and the run is aborted.

If Turbine Release is not user input or set by rules, it is calculated as the minimum of MaxTurbine Release and Outflow minus Spill,

$$TurbineRelease = Min(MaxTurbineRelease, Outflow - Spill)$$

where the Max Turbine Release is interpolated from the Peak Release vs. Net Head Table, given the Net Head over the entire timestep.

In order to calculate the time at peak production, the flow which passes through the turbines during this time period must be calculated. This Peak Release is the maximum possible flow through the turbines given the Net Head and will be solved for iteratively as described in the steps below.

1. Peak Release is initially set to zero.
2. Tailwater Elevation is determined using Peak Release + Spill as the "flow" value in the selected Tailwater method. (If the Turbine Release slot is linked, it can be assumed that Spill is sent elsewhere and does not affect Tailwater so the "flow" value should be set to Peak Release only.)
3. The operating head is calculated as the Pool Elevation minus the Tailwater Elevation.
4. The Net Head is calculated as the operating head minus the head loss.
5. Given the Net Head, the Peak Release is interpolated from the Net Head vs. Peak Release table and then multiplied by the Power Plant Cap Fraction.

6. The new Peak Release value is compared with the previous value and the iteration will continue until the value converges. (Note: Convergence Percentage is a general slot on power reservoirs representing the convergence in all iterative solutions-- the slot defaults to 0.0001 if not input.)

Once Peak Release is calculated, the Peak Time will be solved for as the volume of flow that passes through the turbines in a timestep divided by the Peak Release:

$$\text{Peak Time} = \frac{\text{Timestep Flow Volume}}{\text{Peak Relase}}$$

where timestepFlowVolume is an internal variable defined as:

$$\text{Timestep Flow Volume} = \text{Turbine Release} \times \text{timestep seconds}$$

RiverWare checks that the Peak Time is not greater than the timestep length. If it is, the run aborts and an error is posted.

Power is calculated with the standard water power equation. The Peak Power Equation method uses Peak Release as the flow value and Net Head at Peak Release as the head value.

$$Power = \frac{\text{Peak Release} \times \text{Net Head} \times \text{efficiency} \times \text{Plant Cap Fraction}}{\text{Unit Compatibility Factor}}$$

The unit compatibility factor comes from balancing units and is 102.01697767 in internal RiverWare units.

Energy is finally computed as the product of Power and Peak Time:

$$Energy = Power \times \text{Peak Time}$$

23.1.1.10 Peak Power Equation with Off Peak Spill

The Peak Power Equation with Off Peak Spill method provides a standard equation to calculate peaking power for a portion of the computational timestep using the water power equation. Included also is a calculation of the off peak spill that occurs when the turbines are not operating.

Slots Specific to this Method

☛ Head Loss

Type: Table Slot
Units: LENGTH
Description: The head loss water incurs before it reaches the turbines.
Information: The slot is set to zero if not input by the user.
I/O: optional
Links: Not linkable

☛ Maximum Power Pool Drawdown

Type: Scalar
Units: Length
Description: maximum vertical drop permitted in the power pool in one timestep for power release
Information:
I/O: Required input
Links: Not Linkable

☛ Minimum Power Elevation

Type: Table Slot
Units: length
Description: Minimum pool elevation at which power can be generated.
Information: Single value in a 1x1 table slot
I/O: Required input
Links: Not linkable

☛ Minimum Elevation for Power Operations

Type: Series Slot with Periodic Input, HERE (Slots.pdf, Section 4.8)
Units: Length
Description: Minimum pool elevation at which power operations can occur.
Information: This slot provides another way to limit the additional proposed hydropower release described HERE (USACE_SWD.pdf, Section 3.9). In simulation, a warning is issued if the Pool Elevation is below this elevation.
I/O: Optional input as either series or periodic values.
Links: Not linkable

☛ Net Head vs Plant Efficiency

Type: Table Slot
Units: Length vs none
Description: relationship between the Net Head and the efficiency of the plant
Information: Net Head includes head loss and Efficiency includes both generator and turbine efficiency
I/O: Required Input
Links: Not Linkable

☛ Net Head vs. Generator Capacity

Type: Table Slot
Units: Length vs. Power
Description: relationship between Net Head and the maximum possible power produced
Information: Net Head includes head loss

I/O: Required Input

Links: Not Linkable

☛ Off Peak Spill

Type: Series Slot
Units: Flow
Description: The spill that occurs during the off peak portion of the timestep.
Information: Off Peak Spill is the fraction of the spill that occurs when power is not being produced. The time weighted average of Off Peak Spill and Peak Spill equals the Spill. If the Peak Time equals the timestep length, Off Peak Spill is NaN.
I/O: Output only

Links: Not linkable

☛ Peak Release

Type: Series Slot
Units: FLOW
Description: The flow through the turbines when the plant is operating at generator capacity
Information: Peak Release is solved for iteratively using net head, tailwater elevation and pool elevation
I/O: Output only

Links: Not linkable

☛ Peak Time

Type: Series Slot
Units: TIME
Description: The time at which the plant is operating at peak capacity
Information: Peak Time is calculated as the timestep flow volume divided by the Peak Release.
I/O: Output

Links: Not linkable

☛ Plant Power Limit

Type: Series Slot
Units: Power
Description: The user specified upper limit on power production.
Information: If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit and the Energy is recalculated. A new Turbine Release is then calculated based on the Plant Power Limit.
I/O: Optional Input

Links: Not Linkable

☛ Peak Spill

Type: Series Slot
Units: Flow
Description: The spill that occurs during the Peak Time.
Information: Peak Spill is the portion of the Spill that occurs during the Peak Time. The time weighted average of Peak Spill and Off Peak Spill equals the Spill.
I/O: Output
Links: Not linkable

☛ Power Plant Cap Fraction

Type: Series Slot
Units: NONE
Description: decimal fraction of the power capacity at which the plant is operating
Information: Used in the case of outages or reductions in the plant operating capacity.
I/O: Defaults to 1.0 if not user input. Must be between 0 and 1.
Links: Not linkable

This method is called from the dispatch method, typically after Outflow, Storage, and Pool Elevation have been calculated. If Energy is input, then this method is also called from an iterative loop used to determine the Turbine Release, Peak Release, and Peak Time and/or Spill that satisfies the Energy.

The Peak Power Equation with Off Peak Spill method performs a series of checks. First, the method checks if maximum drawdown is exceeded for two cases. In the first case, if there is a valid Top of Conservation Pool slot, i.e. this is a U.S. Army Corp of Engineers model, the method checks if the Pool Elevation is greater than the Top of Conservation Pool. If so, no error or warning is posted; the drawdown limitation only applies to the conservation pool. If the drawdown is exceeded and the Pool Elevation is less than the top of conservation pool, an error is posted and the run is aborted. Note, an abortive error is only posted if the reservoir is dispatching at the current controller timestep. If it is dispatching at a forecast timestep and max power pool drawdown is exceeded, no error is issued. It is assumed that either the inflow or outflow will be modified when the reservoir dispatches at the current timestep and would catch any errors then.

In the second case, if there is no Top of Conservation Pool slot (i.e. a non U.S. Army Corp of Engineers model), if the calculated Pool Elevation results in exceeding the Maximum Power Pool Drawdown, a warning is posted.

If the calculated Pool Elevation is less than the value in the Minimum Elevation for Power Operations, a warning message is posted.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the Minimum Power Elevation was input by the user. If no value was input, a **RiverWare**™ error is posted and the simulation run is aborted. If the previous Pool Elevation is less than the Minimum Power Elevation or the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Power, Energy, Peak Release, and Peak Time are set equal to zero.
2. If the Turbine Release is input or already set from the Dispatch Method "solveMB_givenInflowRelease," a **RiverWare**™ error is flagged and the run is aborted.
3. Turbine Release is set equal to zero.
4. Tailwater and operating head are computed. No further calculations are performed.

The method checks whether Turbine Release (the average flow through the turbines over the whole timestep) is user input or set by rules. (i.e. the method checks if the dispatch type is solveMB_givenInflowRelease.) If so, and if the given Turbine Release value is greater than Outflow minus Spill, an error is posted and the run is aborted.

In order to calculate the time at peak production, the flow which passes through the turbines during this time period must be calculated. This Peak Release is the maximum possible flow through the turbines given the Net Head and will be solved for iteratively as described in the steps below.

1. Peak Release is initially set to zero.
2. Given the net head from the previous timestep (Operating Head at previous timestep minus Head Loss), the efficiency is interpolated from the Net Head vs Efficiency table. The previous Operating Head is used as an approximation so as not to introduce an additional variable in the iteration. As a result, the Tailwater Elevation at the initial timestep must be input. The net head for the initial timestep is the initial Pool Elevation minus the initial Tailwater Elevation minus Head Loss.
3. The current Tailwater Elevation is determined using the maximum of Peak Release plus Unregulated Spill or the current Outflow as the flow value in the selected Tailwater method. (Note that if Energy is input, the Tailwater Elevation slot value shown will be calculated using the average Outflow from the timestep, but the tailwater elevation used in the Peak Release calculation is calculated as described here. If Energy is not input, the Tailwater Elevation slot value will be the peak Tailwater Elevation calculated here.)
4. The Operating Head is calculated as the average Pool Elevation minus the Tailwater Elevation.

5. The Net Head is calculated as the Operating Head minus the Head Loss.

6. Given the Net Head, the Generator Capacity is interpolated from the Net Head vs. Generator Capacity table and is then multiplied by the Power Plant Cap Fraction. If this new capacity is greater than the Plant Power Limit, if valid, the generator capacity is reset to the Plant Power Limit.

7. Peak Release is calculated according to the power equation. The unit compatibility factor comes from balancing units and the specific weight of water; it is 102.01697767 in internal RiverWare units.

$$\text{Peak Release} = \frac{\text{Generator Capacity} \times \text{Unit Compatibility Factor}}{\text{Net Head} \times \text{Efficiency}}$$

The new Peak Release value is compared with the previous value and the iteration, steps 3-7, continue until the value converges. (Note: Convergence Percentage is a general slot on power reservoirs representing the convergence in all iterative solutions-- the slot defaults to 0.0001 if not input.)

If Turbine Release is not user input or set by rules (not in the dispatch method solveMB_givenInflowRelease), TempTurbineRelease is calculated as the minimum of the Peak Release and Outflow minus Spill,

$$TempTurbineRelease = Min(PeakRelease, Outflow - Spill)$$

The Spill will be non-zero only if there is Unregulated Spill or a spill value is set by user input or rules. Once Peak Release is calculated, the Peak Time will be solved for as the volume of flow that passes through the turbines in a timestep divided by the Peak Release:

$$\text{Peak Time} = \frac{\text{TempTurbine Release} \times \text{timestep seconds}}{\text{Peak Release}}$$

RiverWare checks that the Peak Time is not greater than the timestep length. If it is, the run aborts and an error is posted. Next power is set to be the Generator Capacity:

$$Power = GeneratorCapacity$$

Turbine Release is the Peak Release averaged over the whole timestep:

$$TurbineRelease = \frac{PeakRelease \times PeakTime}{TimestepSeconds}$$

Energy is computed as the product of Power and Peak Time:

$$Energy = Power \times \text{Peak Time}$$

Peak Spill and Off Peak Spill are then determined based on Peak Time, Spill, and Unregulated Spill. If Unregulated Spill is non-zero, then Peak Spill is assumed to be equal to Unregulated Spill. Unregulated spill is calculated based on the pool elevation and occurs over the entire timestep. Off Peak Spill is the sum of Unregulated Spill and the Regulated Spill plus Bypass apportioned over the off peak time. If there is no Unregulated Spill, the

Peak Spill is zero, and the Off Peak Spill is the Regulated Spill plus Bypass apportioned over the off peak time. If the Peak Time is equal to the timestep length, then Peak Spill is equal to Spill and Off Peak Spill remains NaN.

Finally, if the Load slot is visible and valid, the Thermal Purchase, Dump Energy and Operation Factor are calculated. See the Load Calculation Section on page 624 for more information on these slots.

23.1.1.1.11 LCR Power

The LCR Power method uses an empirical relationship to calculate the energy produced by the Hoover, Davis, and Parker dams on the Lower Colorado River. The method replicates the calculations from the U.S. Bureau of Reclamation BHOPS FORTRAN program. Energy is calculated as a function of flow, Operating Head, Plant Efficiency, and the Power Coefficients.

Slots Specific to this Method

☛ LCR Input Efficiency

Type: Series Slot
Units: NONE
Description: a fractional value ranging from 0 to 1 which may be used to scale the efficiency or the turbine units in the hydropower plant.
Information:
I/O: Optional; the value defaults to 1 if not input by the user.
Links: Not linkable

☛ Lower Colo Power Coeffs

Type: Table Slot
Units: NONE
Description: two values used as empirical coefficients in relating flow, head, and efficiency to energy.
Information: The coefficients for Hoover Dam are empirically derived. For Davis and Parker dams, they reduce to coeff1=1 and coeff2=0.
I/O: Required input
Links: Not linkable

☛ Net Energy Request

Type: Series Slot
Units: ENERGY
Description: represents the total energy requested by the grid
Information: This slot is only used when a Best Efficiency flag is set in the Energy slot.
I/O: Optional; only used when a Best Efficiency flag is set in the Energy slot.

Links: Not linkable

☛ PLANT EFFICIENCY

Type: Series Slot
Units: NONE
Description: a fractional value ranging from 0 to 1 that represents the percentage of full efficiency of the turbine units in the hydropower plant
Information: In the case of Davis and Parker dams, this equals the LCR Input Efficiency. For Hoover Dam, a Plant Efficiency is calculated.
I/O: Output only
Links: Not linkable

☛ POWER COEFFICIENT

Type: Series Slot
Units: POWER PER FLOW
Description: power generated per unit power release
Information: This coefficient corresponds to the efficiency of the entire plant. It is not used in calculation and is displayed only for the benefit of the user.
I/O: Output only
Links: Not linkable

☛ STATION ENERGY TABLE

Type: Table Slot
Units: ENERGY
Description: represents the energy required to run the station for each day of the week
Information: This slot is only used when a Best Efficiency flag is set in the Energy slot.
I/O: Optional; only used when a Best Efficiency flag is set in the Energy slot.
Links: Not linkable

The first step in the LCR Power method is to determine the Operating Head. This is accomplished by executing the Tailwater method specified by the user.

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Power, Energy, Plant Efficiency, and Power Coefficient are set equal to zero.

2. If the Turbine Release is input or already set from the Dispatch Method "solveMB_givenInflowRelease," a **RiverWare**™ error is flagged and the run is aborted.

3. Turbine Release is set equal to zero. No further calculations are performed.

If Energy is input by the user or it has been flagged as Best Efficiency, the LCR Power Release method is called to calculate the Turbine Release. This method takes a power request, determines if it can be met given the maximum power that can be generated for the given head, and sets the Turbine Release required to generate the requested power. If the Turbine Release is not already calculated by the LCR Power Release method, it is set as Outflow minus Spill. The value of Turbine Release is checked against the maximum value set by the user (if it has been set). If Turbine Release is greater than the maximum value an error is posted which reads, “Turbine Release required to meet Energy request is greater than the maximum Turbine Release.”

Power and Energy are then calculated by the following equations:

$$\text{Energy (MWH)} = \left(\text{Lower Colo Power Coeffs\_1} \times \frac{62.4}{737.5} \times \text{Outflow (1000 cfs)} \times \text{Timestep (hours)} \times \frac{\text{Operating Head (ft)}}{1000} - \text{Lower Colo Power Coeffs\_2}\right) \times \text{LCR Input Efficiency} \times 1000$$

where 62.4 is the unit weight of water in pounds per cubic foot and 737.5 represents ft.-lb./sec. per Kilowatt.

$$\text{Power} = \frac{\text{Energy}}{\text{Timestep (hours)}}$$

If energy is zero, Plant Efficiency and the Power Coefficient are also zero. Otherwise they are calculated as:

$$\text{Plant Efficiency} = \frac{\text{Energy} / 1000}{\frac{62.4}{737.5} \times \text{Outflow (1000 cfs)} \times \text{Timestep (hours)} \times \frac{\text{Operating Head (ft)}}{1000}}$$

$$\text{Power Coefficient} = \frac{\text{Energy}}{\text{Turbine Release} \times \text{Timestep (hours)}}$$

23.1.1.12 Unit Power Table

This method uses a 3-D table that contains the columns Operating Head, Turbine Release, and Power for **each unit** in the plant.

SLOTS ADDED BY THIS METHOD

Note, many of these slots have column or row dimensions based on the number of units. The rows/columns of these slots are expanded at the beginning of the run to match the value in the Number of Units slot. When first configuring this method, the user must enter the Number of Units, then run the model (stepping through 1 timestep is enough) to grow the slots to the right dimensions.

The following slots are instantiated when this method is selected.

☛ Auto Unit Best Turbine Q Table

Type: Table Slot
Units: Length, flow
Description: Table showing most efficient release levels for given operating heads
Information: This table is generated from the Unit Power Table. The first column for each block is Operating Head and second column is Turbine Release. It will have one block for each unit.
I/O: Automatically generated at beginning of run
Links: Not Linkable

☛ Auto Unit Max Turbine Q Table

Type: Table Slot
Units: Length, Flow
Description: Table showing maximum release possible for given operating heads
Information: This table is generated from the Unit Power Table. The first column for each block is Operating Head and the second is Turbine Release. It will have one block for each unit.
I/O: Automatically generated
Links: Not Linkable

☛ Minimum Power Elevation

Type: Table Slot
Units: Length
Description: The minimum elevation at which the reservoir can still produce power.
Information:
I/O: Optional Input Only
Links: Not Linkable

☛ Number of Units

Type: Table Slot
Units: None
Description: Number of units in the plant
Information: This key scalar slot (existing slot, 1x1 table) indicates the number of units (turbines) at a power reservoir. The dimensions of several other slots are directly related to this value; specifically, both input data represented in Table Slots and unit-level series data represented in Agg Series Slots are require one row or column for each unit. At the **beginning of each run**, RiverWare will confirm that the value for the Number of Units slots is consistent with the dimensions of related slots. If any inconsistencies are detected, the relevant slots are resized as appropriate. If additional input data are required, the user is notified and the run is aborted.

I/O: Required Input only

Links: NA

☛ Number of Units Generating

Type: Series Slot
Units: NONE
Description: Number of units that are generating at a given timestep
Information: The value is the sum of the Unit Is Generating
I/O: Output only
Links: NA

☛ Power Curvature Tolerance

Type: Scalar Slot
Units: NONE
Description: The power curvature tolerance is used to account for anomalies in Unit Power Table data and round off error while calculating slopes.
Information: Although the units for the slot are “None”, the comparison is implicitly using (MW/cms)
I/O: Input or defaults to 1X10-6
Links: Links: Not linkable

☛ Unit Is Generating

Type: Agg Series Slot
Units: NONE
Description: This slot is used to control whether units are available.
Information: There is one column for each unit. Before a run, an input value of 1 indicates that the unit **must** generate power at that date; otherwise an input value of 0 indicates that the unit can **not** generate power at that timestep. A NaN indicates that the unit is available and the model will decide if it can generate. At the end of a run, an output 1 indicates the unit generated power, a 0 indicates it did not.
I/O: Can be input by user
Links: Not Linkable

☛ Unit Energy

Type: Agg Series Slot
Units: ENERGY
Description: Energy produced by each unit
Information: There is one column for each unit. A value indicates the energy being generated by the unit at that timestep, and takes into account frequency

regulation. A negative value can be input or set by a rule to represent a unit that is spinning, motoring, or condensing (actually consuming energy).

I/O: Input, Rules, or Output

Links: Not linkable

☛ Unit Power

Type: Agg Series Slot

Units: Power

Description: The power that is generated by each unit

Information: There will be one column for each unit. A value indicates the power being generated by the unit at that timestep, and takes into account losses due to frequency regulation.

I/O: Calculated

Links: Not linkable

☛ Unit Power Table

Type: Table Slot

Units: Length, Flow, Power, Flow, Power, etc...

Description: A 3 dimensional table relating operating head, turbine release, and power for each unit in the plant. There will be 1 block (3 columns) for each unit.

Information: The last row for each operating head represents the **max capacity**. Best efficiency is automatically calculated. The three values in a given row and unit block represent a legal operating point for that unit, i.e., the Power which that unit would generate at that head and turbine flow. It will be necessary to enforce that a point of zero flow and zero power production be entered in the table for each operating head. It is also required that this table be concave.

I/O: Required Input only

Links: NA

Unit 1			Unit 2		
Operating Head (ft)	Turbine Flow (1000 cfs)	Power (KW)	Operating Head (ft)	Turbine Flow (1000 cfs)	Power (KW)
100	0	0	99	0	0
100	10	2000	99	10	1000
100	20	3000	99	20	2000
100	30	4000	99	30	3000
200	0	0	200	0	0
200	10	2500	200	10	1700
200	20	3500	200	20	2500

Unit 1			Unit 2		
Operating Head (ft)	Turbine Flow (1000 cfs)	Power (KW)	Operating Head (ft)	Turbine Flow (1000 cfs)	Power (KW)
200	25	3800	200	25	2800
200	30	4500	200	30	3500
300	0	0	295	0	0
300	10	3000	295	10	3000
300	25	5000	295	25	4000

Unit Priority Table

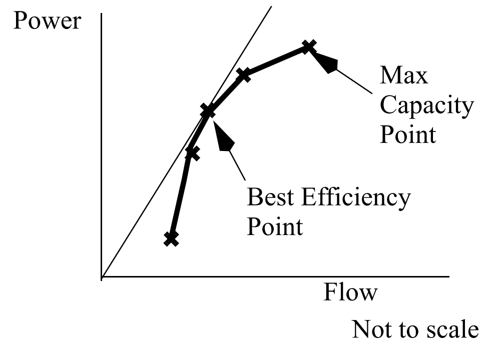
Type: Table Slot
Units: No Units
Description: The priority that each unit is started or stopped in the power plant
Information: There will be one row for each unit. In optimization only, units with lower numerical values are higher priority and are scheduled to release power in preference to lower priority units. If a value is absent then that unit is given the lowest priority. For units with equal priority, the unit efficiency will determine precedence, i.e. a unit with a higher efficiency will be prioritized higher than other units with the same priority value (note, not implemented yet. Currently, units with equal priority are turned on/off in an arbitrary order.). Currently, this table is only used in optimization. It is not used in simulation.
I/O: Optional input
Links: NA

Unit Turbine Release

Type: Agg Series Slot
Units: Flow
Description: Flow through each unit
Information: There is one column for each unit. The value is the expected Turbine Release through the unit at that timestep.
I/O: Input, Rules, Output
Links: Not Linkable

Method Details

This method will use table interpolation to calculate Power and Energy at a known Operating Head based on the characteristics of each unit. This method, Unit Power Table, works in a similar manner to the current method, Plant Efficiency Curve.



At the beginning of the run, the method creates the Auto Unit Best and Auto Unit Max Turbine Q tables from the Unit Power Table. The Auto Unit Max Turbine Q table are the points from the Unit Power Table that correspond to the largest Turbine Release for a given Operating Head. The Auto Unit Best Turbine Q table is calculated from the Unit Power Table as follows: For each unit and each operating head, the method determines the point (flow, power) that corresponds to a line drawn from the origin and it tangent to the curve. This tangent point is determined by calculating the slope of the line from each point to the origin; the point with the largest slope is the tangent. Then the operating head and turbine flow for this point are added to the Auto Unit Best Turbine Q.

The description of the solution in this section assumes that mass balance has already occurred (i.e. Inflow, Outflow, Storage, Pool Elevation have been calculated) and the method is trying to compute the Energy and Power produced by that Outflow. At the start of this method, there is also an estimate of the Turbine Release calculated as

$$\text{Turbine Release} = \text{Outflow} - (\text{Unregulated Spill} + \text{Regulated Spill} + \text{Bypass})$$

where the spills are either known or estimated based on the current Pool Elevation. This Turbine Release may be reset if it cannot be met. If Energy is input, set by a rule, or flagged **BEST EFFICIENCY** (B) or **MAX CAPACITY** (M) or **UNIT** (U) then Energy and Power are solved using the method **Unit Power Table Release** described in Section 23.1.2.8 on page 1120.

In this description, "t" indicates the current timestep and "u" indicates that the method will do this for each unit.

The method calculates the tailwater and operating head using the selected method and current release and pool elevation.

For each unit, the previous pool elevation will be compared to the unit's minimum power elevation.

if (Pool Elevation[t-1] < Unit Minimum Power Elevation[u])
{
{
Either set the following to zero or make sure that they are zero:

- Unit Turbine Release[t,u],
- Unit Energy[t,u],
- Unit Power[t,u], and

- Unit is Generating[t,u].

  If any of these are non-zero, an error is issued.
  }

}

Next, the selected method in the Power Plant Failure category is executed. This method sets the Power Plant Cap Fraction (the default is 1.0) and checks for plant shutoff/failure.

Then, the method checks if the plant has shutoff/failed (from the above call to the selected Power Plant Failure category method), the following steps are taken:

1. Power, Energy, and Turbine Release are set equal to zero.
2. Either set the following to zero or make sure that they are zero:
   - Unit Turbine Release[t,u],
   - Unit Energy[t,u],
   - Unit Power[t,u], and
   - Unit is Generating[t,u].

   If any of these are non-zero, an error is issued No further computations are performed.

Next for each unit, an estimate of max flow through all the turbines is calculated as follows: estimate a temporary variable maxPowerRelease (flow) using the Auto Unit Max Turbine Q table. This table contains the columns Operating Head and Turbine Capacity. Because Operating Head is known at the current timestep, table interpolation is used to calculate maxPowerRelease for the given average Operating Head.

If Outflow is set to **Max Capacity** flag, set the Unit Turbine Release to the maximum calculated and compute the power produced by those flows.

Otherwise, if Unit Turbine Release is input/rules for any of the units:

If Unit is Generating is set (input/rules) to 0 for a unit that has a Unit Turbine release, issue an error.

If Unit is Generating is set (input/rules) to 1 for a unit that does not have a Unit Turbine release, issue an error.

If Turbine Release is not set by the U flag, check if Turbine Release = Σ Unit Turbine Release[u]. If they do not match and Turbine Release is input/rules, issue an error. Otherwise, if they do not match, the method resets Turbine Release equal to Σ Unit Turbine Release[u].

If Turbine Release does have the U flag, the Turbine Release is set equal to Σ Unit Turbine Release[u].

If Turbine Release is now greater than Σ maxPowerRelease[u], an error is issued as the specified unit turbine releases cannot be met.

If a regulation method is selected, call it here, otherwise, given the known Unit Turbine Releases[u], the method then looks up the unit flow and operating head on the Unit Power Table to determine the power produced by each unit: Unit Power[u].

Finally, Unit Energy[u] = Unit Power[u] * Time (hrs)

The total Power = Σ Unit Power[u] and total Energy = Σ Unit Energy [u]

Else Unit Turbine Release is not input/rules

Exit the method as there is no way to compute energy/power at a unit level. The dispatch method will continue but no power related slots (e.g. Energy, Unit Energy, Power, Unit Power, Unit Turbine Release, Unit Is Generating, Unit Startup, Unit Shutdown, etc) will be set. Turbine Release is set to the minimum of (Outflow - Spill or MaxPowerRelease). This can happen when dispatching given Inflow and Pool Elevation or Storage. and no Turbine Release or Unit Turbine Release is specified.

Finally, the method computes the slot Unit is Generating based on the Unit Turbine Release and Unit Energy. For each unit, if these are non zero, the Unit is Generating is set to 1. If they are zero, Unit is Generating is set to 0. No inputs are overwritten. Then, the Number of Units Generating is computed as the sum over the columns of the Unit is Generating slot.

23.1.2 Power Release

When Energy is specified, the Power Release method is used to calculate the Turbine Release required. If the Energy request can not be met, the user is notified. There is one method per Power method.

The Power Release category is available when any of the Power methods is selected **except** None, Peak Power, Peak and Base, or Peak Power Equation.

23.1.2.1 None

This is the default method in the Power Release category. No calculations are performed in this method. There are no slots specifically associated with this method. If this method is selected for the Power Release category, a **RiverWare**™ error will be posted and the simulation run will be aborted. A viable power release method must be selected when the Power Release category is visible.

SLOTS SPECIFIC TO THIS METHOD

☛ **NONE**

23.1.2.2 Plant Power Coefficient Release

The Plant Power Coefficient Release method calculates Turbine Release using the entire plant characteristics when Energy is specified. The Plant Power Coefficient Release method is only available if the Plant Power Coefficient method is selected in the Power category. Energy must be input for this method to execute. If Energy is flagged as either **MAX CAPACITY** or **BEST EFFICIENCY**, it is considered input. If Energy is flagged as **MAX CAPACITY**, Turbine Release is set to meet the Energy request at the maximum flow rate. If Energy is flagged as **BEST EFFICIENCY**, Turbine Release is set to meet the Energy request at the most efficient flow rate. If Energy is neither flagged as **MAX CAPACITY** nor flagged as **BEST EFFICIENCY**, the Turbine Release is calculated from the Energy request and a Power Coefficient. The Power Coefficient may be input by the user or calculated by **RiverWare**™ from interpolation of the Best and Max Power Coefficient tables.

If Energy is flagged **UNIT VALUES** (U), and error is issued. This flag is only available with the “Unit Power Table Release” method.

SLOTS SPECIFIC TO THIS METHOD

☛ **NONE**

The first step in the Plant Power Coefficient Release algorithm is to set the Power Plant Cap Fraction to 1.0 if it is not already known.

If the Energy slot is flagged as MAX CAPACITY, the following steps are taken:

1. Qtemp, a local variable, is calculated from interpolation of the Max Turbine Q table using the Operating Head.
2. Turbine Release is calculated with the following equation:

$$\text{Turbine Release} = \text{Qtemp} \times \text{Power Plant Cap Fraction}$$

3. PCmax, a local variable, is determined from interpolation of the Max Power Coefficient table using the Operating Head. The Power Coefficient is set as PCmax if it is not input.
4. Power and Energy are then calculated using the following equations:

$$\text{Power} = \text{Turbine Release} \times \text{PCmax}$$

$$\text{Energy} = \text{Power} \times \text{Timestep (in hours)}$$

5. If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit and the Energy is recalculated. A new Power Coefficient and Turbine Release are then calculated based on the Plant Power Limit.

If Energy is flagged as BEST EFFICIENCY, the following steps are taken:

1. Qtemp is calculated from interpolation of the Best Turbine Q table using the Operating Head.
2. Turbine Release is computed using the following equation:

$$\text{Turbine Release} = \text{Qtemp} \times \text{Power Plant Cap Fraction}$$

3. PCbest, a local variable, is determined from interpolation of the Best Power Coefficient table using the Operating Head. The Power Coefficient is set as PC best if it is not input.
4. Power and Energy are then calculated using the following equations:

$$\text{Power} = \text{Turbine Release} \times \text{PCbest}$$

$$\text{Energy} = \text{Power} - \text{Timestep (in hours)}$$

5. If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit, the Energy is recalculated, and the Turbine Release is recalculated as Plant Power Limit / PCbest.

If Energy is not flagged as either MAX CAPACITY or BEST EFFICIENCY and the Power Coefficient is input, the following steps are taken.

1. If the Power Coefficient is less than 0.00000001, a **RiverWare**™ error is posted and the simulation run is aborted.

2. Power is calculated using the following equation:

$$\text{Power} = \text{Energy}/\text{Timestep}$$

where the Timestep is in hours.

3. Qout, a local variable, is calculated with the following equation:

$$\text{Qout} = \text{Power}/\text{Power Coefficient}$$

4. Qtemp, a local variable, is determined by the interpolation of the Max Turbine Q table using the Operating Head.

5. Qmax, a local variable, is computed using the following equation:

$$\text{Qmax} = \text{Qtemp} \times \text{Power Plant Cap Fraction}$$

6. If Qout is greater than Qmax, the largest discharge value in the Max Turbine Q table is found. If this value is greater than or equal to Qout, Turbine Release is set equal to Qmax. If the value is less than Qout, a **RiverWare**™ error is posted and the simulation run is aborted.

7. If Qout is less than or equal to Qmax, Turbine Release is set equal to Qout.

If Energy is not flagged as either MAX CAPACITY or BEST EFFICIENCY and the Power Coefficient is not given, the following steps are taken:

1. Power is calculated using the following equation:

$$\text{Power} = \text{Energy}/\text{Timestep}$$

where the timestep is in hours.

2. The best and max power coefficients are interpolated using the Operating Head and the Best Power Coefficient and the Max Power Coefficient tables, respectively.

3. QbestTemp and QmaxTemp (local variables) are then determined using the Operating Head to interpolate values from the Best Turbine Q and Max Turbine Q tables, respectively.

4. Qbest, a local variable, is computed using the following equation:

$$\text{Qbest} = \text{QbestTemp} \times \text{Power Plant Cap Fraction}$$

5. Qmax, a local variable, is calculated using the following equation:

$$\text{Qmax} = \text{QmaxTemp} \times \text{Power Plant Cap Fraction}$$

6. If Power divided by the best power coefficient is less than or equal to Qbest, Turbine Release is set equal to Power divided by the best power coefficient.

7. If Power divided by the max power coefficient is greater than Qmax, Turbine Release is set equal to the max turbine flow.

8. If neither 3) nor 4) is true, an interpolated value (pcoeffINTERP) is found between the best and max power coefficients based on how close Power is to both the product of Qbest and the best power coefficient, and the product of Qmax and the max power coefficient. The following pair of equations is used to quantitatively determine the pcoeffINTERP value:

$$\text{pcoeffFRACTION} = \frac{(\text{Power} - \text{best power coefficient} \times \text{Qbest})}{(\text{max power coefficient} \times \text{Qmax} - \text{best power coefficient} \times \text{Qbest})}$$

$$\text{pcoeffINTERP} = \text{best power coefficient} + (\text{max power coefficient} - \text{best power coefficient}) \times \text{pcoeffFRACTION}$$

9. The Turbine Release is then calculated with the following equation:

$$\text{Turbine Release} = \text{Power} / \text{pcoeffINTERP}$$

23.1.2.3 Plant Efficiency Curve Release

The Plant Efficiency Curve Release method calculates Turbine Release using the entire plant characteristics when Energy is specified. The Plant Efficiency Curve Release method is only available if the Plant Efficiency Curve method is selected in the Power category. Energy must be input or set by a rule for this method to execute. If Energy is flagged as either **MAX CAPACITY** or **BEST EFFICIENCY**, it is considered input. If Energy is flagged as **MAX CAPACITY**, Turbine Release is set to meet the Energy request at the maximum flow rate. If Energy is flagged as **BEST EFFICIENCY**, Turbine Release is set to meet the Energy request at the most efficient flow rate. If Energy is neither flagged as **MAX CAPACITY** nor flagged as **BEST EFFICIENCY**, the Turbine Release is calculated from the Energy request.

If Energy is flagged **UNIT VALUES** (U), and error is issued. This flag is only available with the “Unit Power Table Release” method.

Slots Specific to this Method

☛ None

The first step in the Plant Efficiency Curve Release algorithm is to set the Power Plant Cap Fraction to 1.0 if it is not already known.

If the Energy slot is flagged as MAX CAPACITY, the following steps are taken:

1. Qtemp, a local variable, is calculated as the maximum release using the Operating Head and the Plant Power Table.
2. Turbine Release is calculated with the following equation:

$$\text{Turbine Release} = \text{Qtemp} \times \text{Power Plant Cap Fraction}$$

3. Power is determined directly from the Plant Power Curve.
4. Energy is calculated as:

$$\text{Energy} = \text{Power} \times \text{Timestep}$$

5. The Power Coefficient is calculated as:

$$PowerCoefficient = (Power)/(TurbineRelease)$$

6. If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit and the Energy is recalculated. A new Power Coefficient and Turbine Release are then calculated based on the Plant Power Limit.

If Energy is flagged as BEST EFFICIENCY, the following steps are taken:

1. Qtemp is computed as the most efficient release given the Operating Head and the Plant Power Table.
2. Turbine Release is computed using the following equation:

$$\text{Turbine Release} = \text{Qtemp} \times \text{Power Plant Cap Fraction}$$

3. Power is determined directly from the Plant Power Curve.
4. Energy is calculated as:

$$\text{Energy} = \text{Power} \times \text{Timestep}$$

5. The Power Coefficient is calculated as:

$$PowerCoefficient = (Power)/(TurbineRelease)$$

6. If the Plant Power Limit is exceeded, Power is reduced to the Plant Power Limit and the Energy is recalculated. A new Power Coefficient and Turbine Release are then calculated based on the Plant Power Limit.

If Energy is not flagged as either MAX CAPACITY or BEST EFFICIENCY and the Power Coefficient is input, the following steps are taken.

1. If the Power Coefficient is less than 0.00000001, a **RiverWare**™ error is posted and the simulation run is aborted.
2. Power is calculated using the following equation:

$$\text{Power} = \text{Energy} / \text{Timestep}$$

3. Turbine Release is calculated as:

$$\text{Turbine Release} = \text{Power} / \text{Power Coefficient}$$

If Energy is not flagged as either MAX CAPACITY or BEST EFFICIENCY and the Power Coefficient is not input, the following steps are taken:

1. Power is calculated using the following equation:

$$\text{Power} = \text{Energy} / \text{Timestep}$$

2. The max Turbine Release and Power production are found for the current operating conditions.
3. If input Power is greater than the max Power for current operating conditions, and INPUT_ENERGY _ADJUST method is chosen, Turbine Release is set equal to the max Turbine Release from 2, and Power is set equal to Power from 2. The Power Coefficient is then computed as Power divided by Turbine Release.
4. Otherwise, Turbine Release is found using the Plant Power Table and the Power Coefficient is set as Power divided by Turbine Release.
5. If the Plant Power Limit is exceeded, an error is posted.

Notes on Power Plant Cap Fraction

If the Power Plant Cap Fraction is input by the user, it is necessary for the Plant Power Table to basically be scaled back to account for the operating points when the turbines are operating at less than 100%. To do this, when Turbine Release is known and Power is to be found using the Plant Power Curve, Turbine Release is divided by the Power Plant Cap Fraction. This point is then found in the Plant Power Curve for the current operating head and the Power is found using 3-D interpolation. Finally the Power is multiplied by the Power Plant Cap Fraction to get the actual Power produced for the current timestep.

If Power is known, and Turbine release is to be found in the table. Power is multiplied by the Power Plant Cap Fraction and then this point is found in the Plant Power Curve to solve for Turbine Release. Turbine Release is then divided by the Power Plant Cap Fraction to get the actual Turbine Release for the current timestep.

23.1.2.4 Plant Power Equation Release

The Plant Power Equation Release method calculates Turbine Release using the water power equation when Energy is specified. The Plant Power Equation Release method is only available if the Plant Power Equation method is selected in the Power category. Energy must be input for this method to execute. If Energy is flagged as either **MAX CAPACITY** or **BEST EFFICIENCY**, it is considered input. If Energy is flagged as **MAX CAPACITY**, Turbine Release is set to meet the Energy request at the maximum possible turbine release. If Energy is flagged as **BEST EFFICIENCY**, the run aborts because **BEST EFFICIENCY** is not supported in this method.

If Energy is flagged **UNIT VALUES** (U), and error is issued. This flag is only available with the “Unit Power Table Release” method.

SLOTS SPECIFIC TO THIS METHOD

☛ **NONE**

This method first checks to see if Turbine Release is user input or set by a rule. If it is, the run aborts because both Energy and Turbine Release cannot be input.

If the Energy slot is flagged as MAX CAPACITY, the following steps are taken:

1. Set Turbine Release to be the maximum turbine release calculated by interpolating the Net Head on the Net Head Vs Max Turbine Release table.

2. Once efficiency, Plant Cap Fraction, Net Head, and Turbine Release are all known, Power is solved for using the Power Equation. The unit compatibility factor comes from balancing units and is 102.01697767 in internal RiverWare units.

$$Power = \frac{\text{Turbine Release} \times \text{Net Head} \times \text{efficiency} \times \text{Plant Cap Fraction}}{\text{Unit Compatibility Factor}}$$

If the computed Power is greater than the Plant Power Limit, the Power is reset to the Plant Power Limit. In this case, Turbine Release is re-computed using the above equation rearranged.

3. Lastly, Energy is computed as Power multiplied by the length of the timestep.

$$Energy = \text{Power} \times \text{Length of Timestep}$$

If the Energy slot is not flagged MAX CAPACITY, the following steps are taken:

When the Energy value is known (rather than flagged **Max Capacity**), the Plant Power Equation Release method uses Energy to solve for Power and Turbine Release. Power is simply Energy divided by the length of the timestep:

$$Power = \frac{Energy}{\text{Length of Timestep}}$$

Using Power, the Net Head and Turbine Release are solved for iteratively as described below:

1. If the computed Power is greater than the Plant Power Limit, the specified energy is too large. The selected method in the Input Energy adjustment category is executed. The Reduce Input Energy method reduces the energy to the maximum possible. If the None method is selected, an error will be issued that the specified energy leads to a power that is above the Plant Power Limit.

2. Turbine Release is initially assumed zero

3. Tailwater Elevation is determined via the selected Tailwater method (the “flow” variable is set to Outflow. If Turbine Release is linked it can be assumed that the Turbine Release and Spill are separated and the “flow” variable should be set to Turbine Release.)

4. Operating Head is calculated as Pool Elevation minus Tailwater Elevation

5. Net Head is calculated as Operating Head minus Head Loss

6. Turbine Release is calculated again using the Water Power equation:

$$TurbineRelease = \frac{Power \times \text{Unit Compatibility Factor}}{\text{Net Head} \times \text{Plant Efficiency}}$$

7. The calculated Turbine Release is compared to the initial Turbine Release and the process iterates until the values converge. (Note: Convergence Percentage is a general slot on power reservoirs representing the convergence in all iterative solutions-- the slot defaults to 0.0001 if not user input.)

Once converged, the Net Head is looked up on the Net Head Vs Max Turbine Release table to get the max release. If the Turbine Release is larger than the max release times the Power Plant Cap Fraction, the selected method in the Input Energy adjustment category is executed. The Reduce Input Energy method reduces the energy to the maximum possible. Otherwise, there is too much flow and an error will be issued that the energy request cannot be met.

23.1.2.5 Peak Power Equation with Off Peak Spill Release

The Peak Power Equation with Off Peak Spill Release method calculates the necessary Turbine Release, Peak Release and Peak Time using the water power equation when Energy is specified. The method is only available if the Peak Power Equation with Off Peak Spill method is selected in the Power category. Energy must be input or set by a rule for this method to execute.

Slots Specific to this Method

☛ NONE

This method first checks to see if Turbine Release is user input or set by a rule. If it is, the run aborts because both Energy and Turbine Release cannot be input. When the Energy value is known, the Peak Power Equation with Off Peak Spill Release method uses Energy to solve for Turbine Release, Peak Release and Peak Time as described below:

1. Peak Release is initially set to zero.
2. Given the net head from the previous timestep (Operating Head at previous timestep minus Head Loss), the efficiency is interpolated from the Net Head vs Efficiency table. The previous Operating Head is used as an approximation so as not to introduce an additional variable in the iteration. As a result, the Tailwater Elevation at the initial timestep must be input. The net head for the initial timestep is the initial Pool Elevation minus the initial Tailwater Elevation minus Head Loss.
3. The current Tailwater Elevation is determined using the maximum of Peak Release or the current Outflow as the value in the selected Tailwater method.
4. The Operating Head is calculated as the average Pool Elevation minus the Tailwater Elevation.
5. The net head is calculated as the Operating Head minus the Head Loss.
6. Given the net head, the Generator Capacity is interpolated from the Net Head vs. Generator Capacity table. If the capacity is above the Plant Power Limit, the Generator Capacity is reset to the Plant Power Limit.
7. Peak Release is calculated according to the power equation. The unit compatibility factor comes from balancing units and the specific weight of water; it is 102.01697767 in internal RiverWare units.

$$\text{Peak Release} = \frac{\text{Generator Capacity} \times \text{Unit Compatibility Factor}}{\text{Net Head} \times \text{Efficiency}}$$

8. The new Peak Release value is compared with the previous value and the iteration (steps 3-7) continues until the value converges. (Note: Convergence Percentage is a general slot on power reservoirs representing the convergence in all iterative solutions-- the slot defaults to 0.0001 if not input.)

Power is set equal to the Generator Capacity and Peak Time is:

$$\text{Peak Time} = \frac{Energy}{Power}$$

Turbine Release is the Peak Release average over the timestep:

$$\text{Turbine Release} = \frac{\text{Peak Release} \times \text{Peak Time}}{\text{Timestep Length}}$$

23.1.2.6 Unit Generator Power Release

The Unit Generator Power Release method is only available when Unit Generator Power is selected in the Power category. It is used to calculate the Turbine Release required to produce a given amount of Power. Energy must be input by the user for this method to execute. There are no slots specifically associated with this method.

If Energy is flagged **UNIT VALUES** (U), and error is issued. This flag is only available with the “Unit Power Table Release” method.

The Unit Generator Power Release method begins by computing the availability and power limits of each unit type. Availability and power limit values are computed as the sum of the values from the availability and power limit columns, respectively, in the Generators Available and Limit slot. A value for availability and power limit is computed for each unit type.

The efficiency of each unit type is calculated by the following equation:

$$efficiency = \frac{powerTemp}{flowTemp}$$

PowerTemp and flowTemp, both local variables, are computed from the Best Generator Power and Best Generator Flow tables, respectively, using the current Operating Head. Each unit type is then sorted in descending order based on the computed efficiency.

In order to compute the Turbine Release associated with the known Power, the method begins to add entire unit types (operating according to the best power and flow tables and beginning with the most efficient type) until the Power is exceed or all the unit types have been added. If the Power is exceeded, the last generator type is interpolated to compute the Turbine Release exactly (see equation below). However, if all the unit types have been added and the Power cannot be met, the method assumes all unit types are operating at full capacity (according to the Full Generator Flow and Full Generator Power tables). Then if the Power is exceeded, the last generator type added is interpolated to compute the Power exactly (see equation below). However, if the Power still cannot be met, an error is posted and the run is aborted because the generators are unable to produce the amount of Power specified by the user.

The interpolation equation used to calculate Power is given below:

$$\text{Turbine Release} = \text{oneLessTypeFlow} + \frac{\text{Power} - \text{oneLessTypePower}}{\text{cumulativePower} - \text{oneLessTypePower}} \bullet (\text{cumulativeFlow} - \text{oneLessTypeFlow})$$

where oneLessTypePower is the power produced from all the previous types added (excluding the most recent type added); oneLessTypeFlow is the flow through all the previous unit types (excluding the most recent type added); cumulativePower is the power produced from all the unit types added (including the most recent type); and cumulativeFlow is the flow through all the unit types added (including the most recent type).

Note: The above equation assumes the relationship between power and flow is linear regardless of the actual relationship specified in the power and flow tables. It is also interpolating over an entire type of generators.

23.1.2.7 LCR Power Release

The LCR Power Release method calculates the release from the Lower Colorado River hydropower products. The LCR Power Release method is available only when LCR Power is selected in the Power category. Energy must be input or flagged as **BEST EFFICIENCY** (Energy cannot be flagged **MAX CAPACITY** for the LCR Power method) for this method to execute. It is determined if the requested Power demand can be met. This determination is based on the maximum possible power that can be generated for a given head. If it is possible to meet the requested Power demand, the Turbine Release is set so as to produce the requested Power.

If Energy is flagged **UNIT VALUES** (U), and error is issued. This flag is only available with the “Unit Power Table Release” method.

Slots Specific to this Method

None

The first step in this method is making sure the Lower Colo Power Coeffs are known. If either of these coefficients are not known, a **RiverWare**™ error is flagged and the simulation run is aborted. Then, the LCR Input Efficiency slot is checked. is not known, If it is not known, it is assumed to be 100% efficient and the LCR Input Efficiency is set to 1.0.

If Energy is flagged as **BEST EFFICIENCY**, it is calculated as the Net Energy Request plus the value of energy in the Station Energy Table corresponding to the current day of the week.

If Energy is flagged as **MAX CAPACITY**, an error is given. If Energy is not flagged as either **BEST EFFICIENCY** or **MAX CAPACITY**, it must be input by the user.

Turbine Release is calculated using the following equation:

$$\text{Turbine Release} = \left(\frac{\text{Energy}}{\text{LCR Input Efficiency}} + (1000 \times \text{Lower Colo Power Coeff \#2})\right)$$
$$\times \frac{1000}{\text{Lower Colo Power Coeff \#1} \times \text{Timestep} \times 62.4/737.5 \times \text{Operating Head} \times 35.31467 \times 3.28084}$$

where the Timestep is in hours. The constants used in the above equation are to convert the input to **RiverWare**™ standard units.

The previous equation is based on the energy calculation equation solved for Flow and corrected to standard units (see LCR Power method):

$$\text{Energy (1000 MWH)} = \Big(\Big(\text{Lower Colo Power Coeff \#1} \times \frac{62.4}{737.5} \times \text{Flow (Kcfs)} \times \text{Timestep (hours)} \times \frac{\text{Operating Head (ft)}}{1000}\Big) - \text{Lower Colo Power Coeff \#2}\Big) \times \text{LCR Input Efficiency}$$

where flow is in kcfs, Timestep is in hour, and Operating Head is in feet.

The correction factors used in the above equations are presented below:

$$\text{Energy (1000 MWH)} = \text{Energy (MWH)} \times 1000$$

$$\text{Flow (cfs)} = \text{Flow (Kcfs)} / 1000$$

$$\text{Operating Head (m)} = \text{Operating Head (ft)} / 3.28084$$

$$\text{Turbine Release (cms)} = \text{Flow (cfs)} / 35.31467$$

Once Turbine Release is calculated, it is checked against the maximum allowable turbine release. A **RiverWare**™ error is flagged and the simulation run is aborted if Turbine Release exceeds the maximum allowable turbine release.

23.1.2.8 Unit Power Table Release

This method is only available if the Unit Power Table method is selected in the Power category, (See “Unit Power Table Release” on page 1120.). The method **Unit Power Table Release** calculates Turbine Release when Energy is specified. If Energy is flagged as **BEST EFFICIENCY** (B) or **MAX CAPACITY** (M) or **UNIT VALUES** (U), it is considered input.

If Energy is flagged B, the Unit Best Turbine Q table will be used to determine the best efficiency Turbine Release for the current average Operating Head. This assumes that all units are in use unless specified otherwise in the Unit is Generating slot. Power is then found using the Unit Power Table. If Energy is flagged M, the Unit Max Turbine Q table is used to determine the maximum Unit Turbine Release for the current average Operating Head. This point is then found in the Unit Power Table to determine the maximum power that can be produced for this Operating Head. If Energy is flagged U, the method calculates Unit Turbine Release using table interpolation of Unit Energy on the Unit Power Table with the Unit Energy.

If Energy is input but not flagged as B, M, or U and Unit Energy is not input, the method will exit without calculating Unit Energy. If any of the values in Unit Energy are input, it will be used to determine the release and power.

Method Details

This method will be called if Energy is input or set by a rule, which includes being flagged B, M, or U. This method will execute in the following manner.

if (Energy is flagged M)

{

If any of the Unit Energy[u] values are input or set by a rule, issue an error.
For each unit that is available (based on a non-zero value in the Unit is Generating slot), use 2D interpolation of Auto Unit Max Turbine Q table;
Set max release to a temporary local variable, Qmax[u];
Turbine Release is set to Σ Qmax[u];
Once the value for each unit flow at the current average Operating Head is found, the Unit Power[u] produced for that flow can be determined directly from the Unit Power Table.
}
else if (Energy is flagged B)
{
If any of the Unit Energy[u] values are input or set by a rule, issue an error.
For each unit that is available (based on a non-zero value in the Unit is Generating slot), use 2D interpolation of Auto Unit Best Turbine Q table to determine release at B;
Set best release to a temporary local variable, Qbest[u];
Turbine Release is set to Σ Qbest[u];
Again, Unit Power[u] will then be able to be determined directly from the Unit Power Table.
}
else if (Energy is Input/Rules (including U flag) and Unit Energy for any unit is not input/rules)
{
Issue an error; there is no way to calculate Unit Energy from plant values and no way to calculate plant Power without unit information
}
else if (Energy is input/rules (including U flag) and Unit Energy for any unit is input/rules)
{
If Unit is Generating is set (input/rules) to 0 for a unit that has a Unit Energy, issue an error.
If Unit is Generating is set (input/rules) to 1 for a unit that does not have a Unit Energy, issue an error.
If Energy is flagged U, $Energy = \Sigma UnitEnergy[u]$; otherwise, if $Energy \neq \Sigma UnitEnergy[u]$, an error is issued
Next, Unit Power[u] = Unit Energy[u] / time (hrs)
$Power = \Sigma UnitPower[u]$
From this power calculation, the Unit Turbine Release[u] can then be determined using a reverse table lookup of Unit Power[u] in the Unit Power Table. If the Shared Penstock Head Loss method is selected, the solution is iterative as the net operating head is a function of Turbine Release. If Unit Energy[u] is less than zero, the Unit Turbine Release[u] is set to zero. A negative Unit Energy can be set to represent a unit that is spinning but not producing power (i.e. condensing).

Turbine Release = Σ Unit Turbine Release[u]

}

Finally, the method returns to the Unit Power Table method and computes Unit is Generating and Number of Units Generating. See “Unit Power Table” on page 1101..

23.1.3 Power Unit Information

This category is used to provide information on unit information when one of two plant level power methods is selected. This category is only available if the **Plant Efficiency Curve** or **Plant Power Coefficient** methods are selected.. In this category are two methods:

23.1.3.1 None

This is the default, no-action method.

23.1.3.2 Plant Power Table with Units

When selected, the **Plant Power Table with Units** method allows the user to specify the number of units associated with each Turbine Release / Power combination on the Plant Power Table. In addition, at the end of the power method, the Operating Head and Turbine Release are looked up to compute the number of units that are generating.

SLOTS ASSOCIATED WITH THE METHOD:

☛ **PLANT POWER TABLE**

Type: Table Slot
Units: LENGTH, FLOW, POWER, NONE
Description: 3-D table used to determine power using interpolation
Information: Data must be entered into the table in increasing, concave blocks of the same Operating Head for the 3-dimensional table interpolation to work correctly. For every block of the same Operating Head in column 1, Turbine Release should be listed in increasing, concave order in column 2, and the corresponding Power in column 3. The number of units should be increasing in column 4. There should also be a point of zero Turbine Release and zero Power for each operating head. The second to last row for each operating head is the point of best efficiency. The last row for each operating head is the point of maximum Turbine Release and maximum Power production. If there are only two rows for a given operating head, both the **best efficiency** and **max capacity** are equal to the second row. The table shown below is an example of the proper way to formulate the Plant Power Table, with units.

Operating Head	Turbine Release	Power	Number of Units
30	0	0	0
30	100	100	1
30	200	175	2
40	0	0	0
40	100	125	1
40	220	195	2

Operating Head	Turbine Release	Power	Number of Units
50	0	0	0
50	110	147	1
50	250	205	2

I/O: Input Only

Links: Not Linkable

Number of Units Generating

Type: Series
Units: NONE
Description: The number of units generating on this timestep
Information: The value in this slot is computed by looking up the Operating Head and Turbine Release on the Plant Power Table to find the number of units. Note, this computation is a 3D interpolation on the Plant Power Table so there can be a fractional number of units generating, i.e. 1.7.
I/O: Output Only

Links: Not linkable

23.1.4 Input Energy Adjustment

This method category is only available for Plant Power Coefficient Release, Plant Efficiency Curve Release, or Plant Power Equation Release methods in the Power Release category. Its purpose is to adjust input Energy values if they violate a physical constraint.

23.1.4.1 None

This is the default method. It performs no calculations and there are no slots associated with it. The Energy values will not be adjusted if this method is selected.

23.1.4.2 Reduce Input Energy

This method is used to reduce the input Energy value whenever it exceeds the maximum power (due to turbine capacity).

SLOTS SPECIFIC TO THIS METHOD

☛ **REQUESTED ENERGY**

Type: Series Slot
Units: FLOW
Description: The Energy value before being adjusted
Information: This slot is available so that the user can see when an Energy value is adjusted. The value in this slot is the energy value before being adjusted. A value exists in this slot only if the Energy value is adjusted.
I/O: Output only
Links: Not linkable

If the Energy slot value leads to a power that is greater than the maximum reservoir power (due to plant capacity, Plant Power Limit, etc), this method saves the Energy value in the Requested Energy slot. Then, the Maximum Capacity flag is set on the Energy slot. The reservoir is then forced to resolve with the Energy set to Max Capacity (instead of the original, input value). When the reservoir solves the second time, it computes the maximum reservoir Energy and sets this value on the Energy slot. The Maximum Capacity flag remains on the Energy slot for the timestep in question (and will be saved with the model file).

23.1.5 Power Plant Failure

This category is available when any of the power methods are chosen.

23.1.5.1 None

No power plant failure is modeled. If not input, the **Power Plant Cap Fraction**, if used, is set to 1.0.

23.1.5.2 Max Pool, Outflow, Tailwater

During high flow events, certain conditions cause the power plant to fail and no power can be produced. This method model the following conditions:

- Maximum pool elevation
- Maximum tailwater elevation
- Maximum outflow

Each of the above actually has two values, the first, lower value represents the shutoff criteria. The second, higher value represents the failure criteria. If the shutoff criteria is exceeded, then no power can be produced for that timestep, but if the conditions receded below the criteria, then power can again be produced. If the failure criteria is exceeded, the power plant has failed and no power can be produced from that point forward.

For example, a Pool Elevation above the shutoff limit requires the plant to cease generation. However, the power house is not flooded, and when the pool drops back below this limit, the plant can resume generation. In the second case, the pool is above the failure limit and requires the plant to cease generation, but also floods the powerhouse. In this case, even if the pool drops back below the criteria, the plant cannot resume generation, i.e the plant must “fail” for the rest of the simulation.

SLOTS ADDED BY THIS METHOD:

This method will instantiate slots in the following list.

☛ MAX POOL ELEVATION FOR POWER

Type: Table
Units: LENGTH, LENGTH
Description: Elevations at which no power can be produced
Information: This is 1X2 table slot. The first column contains the **shutoff** elevation. The second column contains a higher elevation representing the **failure** elevation.
I/O: Optional input
Links: NA

☛ MAX TAILWATER ELEVATION FOR POWER

Type: Table

Units: LENGTH, LENGTH
Description: Tailwater elevation at which power can no longer be generated
Information: This is 1X2 table slot. The first column contains the **shutoff** tailwater elevation. The second column contains a higher elevation representing the **failure** tailwater elevation. Once the failure elevation is exceeded, the plant has failed and no power can be produced on any subsequent timesteps.
I/O: Optional input
Links: NA

☛ MAX OUTFLOW FOR POWER

Type: Table
Units: FLOW, FLOW
Description: Reservoir outflow at which power can no longer be generated
Information: This is 1X2 table slot. The first column contains the **shutoff** outflow. The second column contains a higher outflow representing the outflow at which the power plant **fails**.
I/O: Optional input
Links: NA

☛ POWER PLANT CAP FRACTION

Type: Series
Units: NO UNITS
Description: This slot tracks whether power production is possible at this timestep
Information: A value of 1 indicates power can be generated, a value of 0 indicates no power can be generated.
I/O: Input or Output
Links: NA

METHOD DETAILS:

This method is executed at the beginning of each power method. First, the failure conditions are checked, then the shutoff conditions are checked.

Failure is tracked using the **Power Plant Cap Fraction.** If **Power Plant Cap Fraction** is 1.0 the power plant is available. If it is 0.0, the plant has failed.

At the beginning of the power method, the following logic determines the **Power Plant Cap Fraction** to use.

If the **Power Plant Cap Fraction** is input, that input value is used and no further checking will be done.

If the previous **Power Plant Cap Fraction** is 0.0, then the current **Power Plant Cap Fraction** is set to zero. This indicates the power plant has failed on previous timesteps and should remain failed. No further checking is done.

Else, the previous **Power Plant Cap Fraction** is unknown or non-zero. Then, if any of the following are true, then the **Power Plant Cap Fraction** is set to 0.0; the plant has failed.

$$\text{Pool Elevation}[t-1] > Max \text{ Elevation for Power}[failure]$$

$$\text{Tailwater Elevation}[t-1] > \text{Max TailwaterElevation for Power}[failure]$$

$$Outflow[t-1] > Max \text{ Outflow for Power}[failure]$$

A diagnostic is available in the User Methods category that describes any failure constraints in effect.

Otherwise, the **Power Plant Cap Fraction** is set to 1.0

Even if the plant "fails" in the course of a simulation, the user can "restart" it manually by setting a non-zero value in the **Power Plant Cap Fraction.**

If any of the following are true, then the plant has failed or shutoff. Turbine Release, Power, Energy and a few method specific slots (as described in the power method section) are set to zero.

$$\text{Power Plant Cap Fraction}[t] = 0.0$$

$$\text{Pool Elevation}[t-1] > Max \text{ Elevation for Power}[shutoff]$$

$$\text{Tailwater Elevation}[t-1] > \text{Max TailwaterElevation for Power}[shutoff]$$

$$Outflow[t-1] > Max \text{ Outflow for Power}[shutoff]$$

$$\text{Pool Elevation}[t-1] < \text{Minimum Power Elevation}$$

Note, the last equation is the behavior for minimum power pool. This slot is added by some of the power methods, not this method.

Also note, if Power Plant Cap Fraction is not zero but one of the other constraints is true, the power slots are set to zero, but the plant has not failed, so future timesteps can generate power.

If none of the above are true, the power method then proceeds as before using the computed **Power Plant Cap Fraction**.

23.1.6 Energy in Storage

The methods available in the Energy in Storage category are used to calculate the total energy that could be produced by the water stored in the reservoir.

23.1.6.1 None

Chosen if the user does not want to calculate the Energy In Storage. No slots are specifically associated with this method. This method performs no calculations.

23.1.6.2 EIS Table Lookup

The EIS TableLookup method obtains the amount of Energy In Storage from a table of Pool Elevation vs. Energy In Storage values and the Pool Elevation.

Slots Specific to this Method

☛ Energy In Storage

Type: AggSeriesSlot
Units: ENERGY
Description: Energy In Storage in the Reservoir
Information:
I/O: Output only
Links: Usually not linked

☛ Energy In Storage Table

Type: TableSlot
Units: LENGTH vs. ENERGY
Description: Pool Elevation vs. Energy In Storage In the Reservoir
Information:
I/O: Required input
Links: Not linkable

The calculations involved with this method are very simple. The Pool Elevation for the current timestep is used to determine the Energy In Storage using the Energy In Storage Table. Simple linear interpolation is used.

23.1.6.3 EIS Table Lookup with Cons Pool

This method will be available only when the **Conservation Pool** or **Conservation and Flood Pools** method in the **Operating Levels** category is selected.

Slots Specific to this Method

☛ Energy In Storage

Type: AggSeriesSlot
Units: ENERGY
Description: Energy In Storage in the Reservoir
Information:
I/O: Output only
Links: Usually not linked

☛ Energy In Storage Table

Type: TableSlot
Units: LENGTH vs. ENERGY
Description: Pool Elevation vs. Energy In Storage In the Reservoir
Information:
I/O: Required input
Links: Not linkable

☛ Conservation Pool Full EIS

Type: Series Slot
Units: MWH
Description: The EIS at the Top of the Conservation Pool
Information:
I/O: Output Only
Links: Not Linkable

The method is executed at the end of each dispatch method.

The **Pool Elevation** for the timestep is looked up in **Energy In Storage Table**. Simple linear interpolation is used. The resulting **Energy in Storage** is then set on the slot.

Next, the **Conservation Pool Full EIS** is calculated as follows:

$$\text{Top Conservation Pool Elevation}[t] = \text{Operating Level Table}[t, \text{Top of Conservation Pool Level}]$$

$$\text{Conservation Pool Full EIS}[t] = \text{Energy in Storage Table}[\text{Top of Conservation Pool Elevation}[t]]$$

Note, in the equation, the appropriate Operating Level Table will be used based on the timestep and the computation in the selected method in the **Conditional Operating Levels** category.

23.1.7 Tailwater

The Tailwater methods (available in the Tailwater category of the User Selectable Methods) calculate the Tailwater Elevation of a Level Power Reservoir. The Tailwater Elevation represents the water surface elevation immediately downstream of the Power Plant. This parameter is required to calculate the Operating Head of the Power Reservoir which is used to calculate either the Energy or Turbine Release of that Reservoir. The Tailwater methods are dependent upon the Reservoir being a Slope Power Reservoir and a valid Power method (available in the Power category of the User Selectable Methods) being selected. These methods require a valid Outflow to perform their calculations.

23.1.7.1 None

This is the default method in the Tailwater category. This method performs no calculations. There are no slots specifically associated with this method.

SLOTS SPECIFIC TO THIS METHOD

☛ **NONE**

23.1.7.2 Linked or Input

The Linked or Input method allows the user to either input values for the Tailwater Elevation or link the Tailwater Base Value to a slot (Pool Elevation or Backwater Elevation) on a downstream Reservoir. If the Tailwater Base Value is linked, the Tailwater Elevation cannot be input by the user. If the Tailwater Base Value is not linked, the Tailwater Elevation must be input by the user. The Tailwater Elevation is determined by following procedure if it is not input (i.e. the Tailwater Base Value is linked).

SLOTS SPECIFIC TO THIS METHOD

☛ **Tailwater Base Value**

Type: SeriesSlot
Units: LENGTH
Description: elevation of tailwater or base elevation used to compute elevation of tailwater
Information:
I/O: Optional; may be input or set by a link.
Links: May be linked to the Pool Elevation or Backwater Elevation of a downstream Reservoir.

The Linked or Input user method does not perform any calculations per se. There are however many logical evaluations performed in this method. The Linked or Input method is discussed below.

If a value for the Tailwater Base Value has been calculated by another user method or propagated through a link, the temporary value for the Tailwater Base Value (TWBaseValueTemp) is set equal to the calculated or linked Tailwater Base Value.

If the Tailwater Base Value is linked to another slot, the following steps are performed:

1. If the Tailwater Elevation is input, a **RiverWare**™ error is flagged and the run is aborted.

2. If TWBaseValueTemp is known, the Tailwater Elevation is set equal to the TWBaseValueTemp.

3. If TWBaseValueTemp is not known, but the previous timestep's Tailwater Elevation is known, the Tailwater Elevation is set equal to the previous timestep's Tailwater Elevation.

4. If neither the TWBaseValueTemp nor the previous timestep's Tailwater Elevation is known and the previous timestep was the initial timestep of the simulation, a **RiverWare**™ error is flagged and the simulation run is aborted.

5. If neither the TWBaseValueTemp nor the previous timestep's Tailwater Elevation is known and the previous timestep was not the initial timestep, the method is exited and waits for more information.

If the Tailwater Base Value is not linked to another slot, Tailwater Elevation must be input. If the Tailwater Elevation is not input, a **RiverWare**™ error is flagged and the simulation run is aborted.

23.1.7.3 Base Value Only

The Base Value Only method is similar to the Linked or Input method. If this method is selected, the user must either input values directly into the Tailwater Base Value slot or link the Tailwater Base Value slot to either the Pool Elevation slot or the Backwater Elevation slot of the downstream Reservoir. Either the initial Tailwater Elevation or the initial Tailwater Base Value must be input by the user for this method to execute successfully. The Tailwater Elevation for any timestep is computed as the average Tailwater Base Value over the timestep.

Slots Specific to this Method

☛ **Tailwater Base Value**

Type: SeriesSlot
Units: LENGTH
Description: elevation of tailwater or base elevation of tailwater is used to compute the tailwater elevation
Information:
I/O: Optional; may be input or linked.
Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

This method is based on logic similar to that of the Linked or Input method. The Base Value Only user method performs many logical evaluations to set the Tailwater Elevation. This method is described below.

The temporary Tailwater Base Value (TWBaseValueTemp) is set equal to the Tailwater Base Value if the Tailwater Base Value was calculated by another user method or set via a link.

If the Tailwater Base Value is linked to another slot, the following steps are performed:

1. If the previous timestep's Tailwater Base Value is not known, and both the previous timestep's Tailwater Elevation and TWBaseValueTemp are known, the Tailwater Elevation is calculated using the following formula:

$$\text{Tailwater Elevation} = \frac{(\text{TWBaseValueTemp} + \text{Tailwater Elevation}(-1))}{2}$$

where: Tailwater Elevation (-1) is the previous timestep's Tailwater Elevation.

2. If neither the previous timestep's Tailwater Base Value nor the TWBaseValueTemp is known but the previous timestep's Tailwater Elevation is known, the current timestep is set equal to the previous timestep's Tailwater Elevation.

3. If neither the previous timestep's Tailwater Base Value nor the previous timestep's Tailwater Elevation is known and the previous timestep is the initial timestep, a **RiverWare**™ error is posted and the simulation run is aborted.

4. If neither the previous timestep's Tailwater Base Value nor the previous timestep's Tailwater Elevation is known and the previous timestep is not the initial timestep, more information must be known. The method is exited and waits for more information to be known.

5. If both the previous timestep's Tailwater Base Value and the TWBaseValueTemp are known, the Tailwater Elevation is calculated using the following equation:

$$\text{Tailwater Elevation} = \frac{(\text{TWBaseValueTemp} + \text{Tailwater Base Value}(-1))}{2}$$

6. If the previous timestep's Tailwater Base Value is known but TWBaseValueTemp is not known, Tailwater Elevation is set equal to the previous timestep's Tailwater Base Value.

If the Tailwater Base Value is not linked, the following steps are performed:

1. If the Tailwater Elevation is not known and the TWBaseValueTemp is known, the Tailwater Elevation is set equal to the TWBaseValueTemp.

2. If neither the Tailwater Elevation nor the TWBaseValueTemp are known, a **RiverWare**™ error is flagged and the simulation run is aborted.

23.1.7.4 Base Value Plus Lookup Table

The Base Value Plus Lookup Table method computes the Tailwater Elevation. This is done by adding the average Tailwater Base Value (over the timestep) to a function of Outflow determined by the Tailwater Table. The Outflow value used to find the corresponding Tailwater value on the Tailwater Table is either the value of the local variable Flow or the value of the Outflow slot. The Tailwater Base Value may be input by the user or linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir. If the Tailwater Base Value is neither input nor linked, it is assumed to be zero.

SLOTS SPECIFIC TO THIS METHOD

☛ **Tailwater Base Value**

Type: SeriesSlot
Units: LENGTH
Description: the base elevation of the tailwater
Information:
I/O: Optional; can be input or linked.
Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

☛ **Tailwater Table**

Type: Table
Units: FLOW vs. LENGTH
Description: reservoir outflow vs. either the tailwater elevation or the tailwater elevation increment
Information: If the Tailwater Base Value is non-zero, the Tailwater Table gives values of incremental increase in Tailwater Elevation over the Base value. Otherwise, the table gives the Tailwater Elevation values.
I/O: Required input
Links: Not linkable

This method is based on logic similar to that of the Linked or Input method. The Base Value Plus Lookup Table user method performs many logical evaluations to set the Tailwater Elevation. This method is described below.

The temporary Tailwater Base Value (TWBaseValueTemp) is set equal to the Tailwater Base Value if the Tailwater Base Value was calculated by another user method or set via a link.

If the Local Info variable Flow is known, the local variable, tempflow, is set equal to Flow. If the Local Info variable Flow is not known, but the Outflow is known, tempflow is set equal to Outflow.

If the tempflow value is known, the following steps are taken to determine the Tailwater Elevation.

If the Tailwater Base Value is linked (and tempflow is known), the following steps are performed:

1. If both TWBaseValueTemp and the previous timestep's Tailwater Base Value are known, TWTemp (a local variable) is obtained from a table interpolation performed

on the Tailwater Table using tempflow. The Tailwater Elevation is then calculated using the following equation.

$$\text{Tailwater Elevation} = (\text{TWBaseValueTemp} + \text{Tailwater Base Value (-1)})/2 + \text{TWTemp}$$

2. If TWBaseValueTemp is known but the previous timestep's Tailwater Base Value is known, but the previous timestep's Tailwater Basic Value is not known. TWTemp is obtained from a table interpolation performed on the Tailwater Table using tempflow. The Tailwater Elevation is then calculated using the following equation:

$$\text{Tailwater Elevation} = \text{TWBaseValueTemp} + \text{TWTemp}$$

3. If TWBaseValueTemp is not known but the previous timestep's Tailwater Base Value is known, TWTemp is obtained from a table interpolation performed on the Tailwater Table using tempflow. The Tailwater Elevation is then calculated using the following equation:

$$\text{Tailwater Elevation} = \text{Tailwater Base Value(-1)} + \text{TWTemp}$$

4. If neither TWBaseValueTemp nor the previous timestep's Tailwater Base Value are known but the previous timestep's Tailwater Elevation is known, the current timestep's Tailwater Elevation is set equal to the previous timestep's Tailwater Elevation.

5. If neither TWBaseValueTemp nor the previous timestep's Tailwater Base Value is known and the previous timestep is the initial timestep of the run, a **RiverWare**™ error is posted and the run is aborted.

6. If neither the TWBaseValueTemp nor the previous timestep's Tailwater Base Value is known and the previous timestep is not the initial timestep of the run, the method is exited and waits for more information.

If the Tailwater Base Value is NOT linked (and tempflow is known), the following steps are performed:

1. If TWBaseValueTemp is known, TWTemp is obtained from a table interpolation performed on the Tailwater Table using tempflow. The Tailwater Elevation is then calculated using the following equation:

$$\text{Tailwater Elevation} = \text{TWBaseValueTemp} + \text{TWTemp}$$

2. If TWBaseValueTemp is not known, the Tailwater Elevation is obtained from a table interpolation performed on the Tailwater Table using tempflow.

The only case where tempflow is not known is when Outflow is not known. This scenario only occurs at the first timestep in one of the three "given Energy" dispatch methods. The following steps are performed to determine the Tailwater Elevation if tempflow is not known.

If the Tailwater Base Value is linked (and tempflow is not known), the following steps are taken:

1. If TWBaseValueTemp and the previous timestep's Tailwater Elevation are known, the Tailwater Elevation is calculated using the following equation:

$$\text{Tailwater Elevation} = (\text{TWBaseValueTemp} + \text{Tailwater Elevation }(-1))/2$$

2. If both TWBaseValueTemp and the previous timestep's Tailwater Base Value are known but the previous timestep's Tailwater Elevation is not known, Tailwater Elevation is calculated using the following equation:

$$\text{Tailwater Elevation} = (\text{TWBaseValue Temp} + \text{Tailwater Base Value }(-1))/2$$

3. If TWBaseValueTemp is known, but neither the previous timestep's Tailwater Elevation nor the previous timestep's Tailwater Base Value are known, Tailwater Elevation is set equal to TWBaseValueTemp.

4. If TWBaseValueTemp is not known but the previous timestep's Tailwater Elevation is known, the current timestep's Tailwater Elevation is set equal to the previous timestep's Tailwater Elevation.

5. If neither TWBaseValueTemp nor the previous timestep's Tailwater Elevation are known, but the previous timestep's Tailwater Base Value is known, Tailwater Elevation is set equal to the previous timestep's Tailwater BaseValue.

6. If the previous timestep is the initial timestep of the run, and none of the following are known: TWBaseValueTemp, the previous timestep's Tailwater Base Value, and the previous timestep's Tailwater Elevation, a **RiverWare**™ error is posted and the simulation run is aborted.

7. If the previous timestep is NOT the initial timestep of the run, and none of the following are known: TWBaseValueTemp, the previous timestep's Tailwater Base Value, and the previous timestep's Tailwater Elevation, the method is exited and waits for more information to execute.

If the Tailwater Base Value is not linked (and tempflow is NOT known), the following procedures are performed:

1. If the Tailwater Elevation is known, the method is exited because no calculations need to be performed.

2. If the Tailwater Elevation is NOT known, and TWBaseValueTemp is NOT equal to zero, Tailwater Elevation is set equal to TWBaseValueTemp.

3. If the Tailwater Elevation is NOT known and either the temporary Tailwater Base Value is NOT known or equal to zero or both, Tailwater Elevation is set equal to the elevation corresponding to zero on the Tailwater Table.

Definitions of some of the terms used in the above equations are located below:

- ***TWBaseValueTemp*** **-** a temporary value for the Tailwater Base Value. This value may be determined from another method or propagated across a link.
- ***tempflow*** **-** a local variable used as the outflow from the Reservoir.
- ***TWTemp*** **-** a local variable used to represent the incremental increase in the Tailwater Elevation over the Tailwater Base Value.
- ***Tailwater Elevation (-1)*** **-** the Tailwater Elevation at the previous timestep.
- ***Tailwater Base Value (-1)*** **-** the Tailwater Base Value at the previous timestep.

23.1.7.5 Stage Flow Lookup Table

The Stage Flow Lookup Table method is similar to the other methods for determining Tailwater Elevation. The Tailwater Elevation is obtained from a 3-dimensional table relating Outflow, Downstream Stage, and the corresponding Tailwater Elevation for most cases. The data in this table must be input by the user. The value for Downstream Stage that is used in this method is the larger of the Tailwater Reference Elevation or the Tailwater Base Value if the Tailwater Base Value is linked to a downstream elevation. If the Tailwater Base Value is not linked, the temporary Tailwater Base Value is used as the Downstream Stage. An average value for the Tailwater Elevation over the timestep is used whenever possible.

Slots Specific to this Method

☛ **Stage Flow Tailwater Table**

Type: Table

Units: FLOW vs. LENGTH vs. LENGTH

Description: Reservoir Outflow vs. Downstream Elevation (Tailwater Base Value) vs. Tailwater Elevation

Information: Data must be entered into the table in increasing blocks of the same Outflow value for the 3-dimensional table interpolator to work correctly. For every block of same Outflows in column 1, Stages should be listed in increasing order in column 2, and the corresponding Tailwater Elevations in column 3.

Outflow	Downstream Stage	TW Elevation
100	500	510
100	550	560
100	600	610
200	500	520
200	550	570
200	600	620

Outflow	Downstream Stage	TW Elevation
300	500	530
300	550	580
300	600	630

I/O: Required input

Links: Not linkable

Tailwater Base Value

Type: SeriesSlot
Units: LENGTH
Description: base elevation of the tailwater
Information:
I/O: Optional; can be input or set by a link.

Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

Tailwater Reference Elevation

Type: Table
Units: LENGTH
Description: lowest Reservoir discharge Elevation when there are no backwater effects from a downstream pool (reservoir)
Information: If this slot has input data, the greater of the Tailwater Reference Elevation or the linked lower reservoir's Pool or Backwater Elevation is used to calculate the Tailwater Base Value. If the Tailwater Base Value is linked to a downstream elevation, this value must be specified by the user. If the Tailwater Base Value is not linked, this value is not used, even if specified.
I/O: Required input

Links: Not linkable

The Stage Flow Lookup Table user method performs many logical tests to determine the Tailwater Elevation. This method is described below.

The first step in this method is to check and see if the Tailwater Reference Elevation is known. If the Tailwater Reference Elevation is not known, a **RiverWare**™ error is posted and the run is aborted.

The temporary Tailwater Base Value (TWBaseValueTemp) is set equal to the Tailwater Base Value if the Tailwater Base Value was calculated by another user method or set via a link.

If the local variable Flow is known, the local variable tempflow is set equal to Flow If the local variable Flow is not known, but the Outflow is known, if the Local Info variable Flow is not known but Outflow is known, tempflow is set equal to Outflow.

If the tempflow value is known, the following steps are taken to determine the Tailwater Elevation.

If the Tailwater Base Value is linked (and tempflow is known), the following steps are performed:

1. If both TWBaseValueTemp and the previous timestep's Tailwater Base Value are known, downstreamStage is computed as the average of the previous timestep's Tailwater Base Value and either the Tail Water Reference Elevation or TWBaseValueTemp (whichever is greater). A table interpolation is performed to determine the Tailwater Elevation using the downstreamStage, tempflow, and the Stage Flow Tailwater Table.

2. If TWBaseValueTemp is known but the previous timestep's Tailwater Base Value is NOT known, downstreamStage is set equal to either the Tail Water Reference Elevation or TWBaseValueTemp (whichever is greater). A table interpolation is performed to determine the Tailwater Elevation using the downstreamStage, tempflow, and the Stage Flow Tailwater Table.

3. If the previous timestep's Tailwater Base Value is known but TWBaseValueTemp is NOT known, downstreamStage is set equal to the previous timestep's Tailwater Base Value. A table interpolation is performed to determine the Tailwater Elevation using the downstreamStage, tempflow, and the Stage Flow Tailwater Table.

4. If neither TWBaseValueTemp nor the previous timestep's Tailwater Base Value are known, but the previous timestep's Tailwater Elevation is known, the current timestep's Tailwater Elevation is set equal to the previous timestep's Tailwater Elevation.

5. If the previous timestep was the initial timestep and none of the following are known: TWBaseValueTemp, the previous timestep's Tailwater Base Value, and the previous timestep's Tailwater Elevation, a **RiverWare**™ error is posted and the run is aborted.

6. If the previous timestep was NOT the initial timestep, and none of the following are known: the temporary TWBaseValueTemp, the previous timestep's Tailwater Base Value, and the previous timestep's Tailwater Elevation, the method is exited and waits for more information.

If the Tailwater Base Value is not linked (and tempflow is known), the following steps are performed:

1. If TWBaseValueTemp is known, downstreamStage is set equal to TWBaseValueTemp. A table interpolation is performed to determine the Tailwater

Elevation using the downstreamStage, tempflow, and the Stage Flow Tailwater Table.

2. If TWBaseValueTemp is NOT known, a **RiverWare**™ error is posted and the run is aborted.

The only case where tempflow is not known is when Outflow is not known. This scenario only occurs at the first timestep in one of the three “given Energy” dispatch methods. The following steps are performed to determine the Tailwater Elevation if tempflow is not known.

If the Tailwater Base Value is linked (and tempflow is not known), the following steps are performed:

1. If both TWBaseValueTemp and the previous timestep’s Tailwater Elevation are known, the Tailwater Elevation is calculated using the following equation:

$$\text{Tailwater Elevation} = (\text{TWBaseValue Temp} + \text{Tailwater Elevation}(-1))/2$$

2. If both TWBaseValueTemp and the previous timestep’s Tailwater Base Value are known but the previous timestep’s Tailwater Elevation is not known, Tailwater Elevation is computed using the following equation:

$$\text{Tailwater Elevation} = (\text{TWBaseValue Temp} + \text{Tailwater Base Value}(-1))/2$$

3. If TWBaseValueTemp is known but neither the previous timestep’s Tailwater Elevation nor the previous timestep’s Tailwater Base Value are known, Tailwater Elevation is set equal to TWBaseValueTemp.

4. If TWBaseValueTemp is not known but the previous timestep’s Tailwater Elevation is known, the current timestep’s Tailwater Elevation is set equal to the previous timestep’s Tailwater Elevation.

5. If neither TWBaseValueTemp nor the previous timestep’s Tailwater Elevation is known, but the previous timestep’s Tailwater Base Value is known, Tailwater Elevation is set equal to the previous timestep’s Tailwater Base Value.

6. If the previous timestep is the initial timestep of the run, and none of the following are known: TWBaseValueTemp, the previous timestep’s Tailwater Base Value, and the previous timestep’s Tailwater Elevation, a **RiverWare**™ error is posted, and the run is aborted.

7. If the previous timestep is NOT the initial timestep of the run and none of the following are known: TWBaseValueTemp, the previous timestep’s Tailwater Base Value, and the previous timestep’s Tailwater Elevation, the method is exited and waits for more information.

If the Tailwater Base Value is not linked (and tempflow is not known) the following procedures are performed:

1. If the Tailwater Elevation is known, the method is exited because no calculations need to be performed.

2. If the Tailwater Elevation is NOT known and the TWBaseValueTemp is NOT equal to zero, Tailwater Elevation is set equal to TWBaseValueTemp.

3. If the Tailwater Elevation is NOT known and either the temporary Tailwater Base Value is NOT known or equal to zero or both, Tailwater Elevation is set equal to the elevation corresponding to zero on the Tailwater Table.

Definitions of some of the terms used in the above equations are located below:

- ***TWBaseValueTemp* -** a temporary value for the Tailwater Base Value. This value may be determined from another method or propagated across a link.
- ***tempflow* -** a local variable used as the outflow from the Reservoir.
- ***downstreamStage* -** a local variable used to hold the value of the downstream stage elevation.
- ***Tailwater Elevation (-1)* -** the Tailwater Elevation at the previous timestep.
- ***Tailwater Base Value (-1)* -** the Tailwater Base Value at the previous timestep.

23.1.7.6 Compare to Avg Base Value

The Compare to Avg Base Value method uses similar methodology as the Base Value Plus Lookup Table and Stage Flow Lookup Table methods. The only difference between this method and the others is that a new local variable, TWCompare, is used. The value for TWCompare is compared with a table lookup value for the Tailwater Elevation. The larger of the two values is used as the Tailwater Elevation. An average Tailwater Elevation over the timestep is calculated whenever possible.

Slots Specific to this Method

Tailwater Base Value

Type: SeriesSlot
Units: LENGTH
Description: base elevation of the tailwater
Information:
I/O: Optional; can be input of set by a link.
Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

☛ **Tailwater Table**

Type: Table
Units: FLOW VS. LENGTH
Description: Reservoir Outflow vs. either the Tailwater Elevation or the tailwater elevation increment
Information: If the Tailwater Base Value is non-zero, the Tailwater Table holds values of incremental increase in Tailwater Elevation over the Base value. Otherwise, the table holds the Tailwater Elevation values.
I/O: Required input
Links: Not linkable

The Compare to Avg Base Value user method performs many logical evaluations to determine the Tailwater Elevation. This method is described below.

The temporary Tailwater Base Value (TWBaseValueTemp) is set equal to the Tailwater Base Value is the Tailwater was calculated by another user method or set via a link. For the Tailwater Compare Method, the Tailwater Base Value must be linked to another object. Otherwise there would be no reason to compare values. If the Tailwater Base Value is not linked, a **RiverWare**™ error is posted and the simulation run is aborted.

If the Local Info variable Flow is known, the local variable tempflow is set equal to Flow.

If the local variable Flow is not known, but Outflow is known, tempflow is set equal to Outflow.

If the tempflow value is known, the following steps are taken to determine the Tailwater Elevation.

1. If both TWBaseValueTemp and the previous timestep's Tailwater Base Value are known,

$$\text{TWCompare} = (\text{TWBaseValueTemp} + \text{Tailwater Base Value}(-1))/2$$

2. If TWBaseValueTemp is known but the previous timestep's Tailwater Base Value is NOT known, TWCompare is set equal to TWBaseValueTemp.

3. If TWBaseValueTemp is not known but the previous timestep's Tailwater Base Value is known, TWCompare is set equal to the previous timestep's Tailwater Base Value.

4. If neither TWBaseValueTemp nor the previous timestep's Tailwater Base Value are known and the previous timestep is the initial timestep, a **RiverWare**™ error is posted and the run is aborted.

5. If neither TWBaseValueTemp nor the previous timestep's Tailwater Base Value are known and the previous timestep is NOT the initial timestep, the method is exited and waits for more information.

Once the TWCompare value has been determined, a table interpolation is performed using the Tailwater Table to obtain the tailwater elevation corresponding to the tempflow value. The tailwater elevation value obtained from this interpolation is then compared to TWCompare. The largest of the two values is used to set the Tailwater Elevation.

The only case where tempflow is not known is when Outflow is not known. This scenario only occurs at the first timestep in one of the three “given Energy” dispatch methods. The following steps are performed to determine the Tailwater Elevation if tempflow is not known.

1. If both TWBaseValueTemp and the previous timestep’s Tailwater Elevation are known,

$$\text{Tailwater Elevation} = (\text{TWBaseValue Temp} + \text{Tailwater Elevation }(-1))/2$$

2. If both TWBaseValueTemp and the previous timestep’s Tailwater Base Value are known but the previous timestep’s Tailwater Elevation is not known,

$$\text{Tailwater Elevation} = (\text{TWBaseValue Temp} + \text{Tailwater Base Value }(-1))/2$$

3. If TWBaseValueTemp is linked and the temporary Tailwater Base Value is known but neither the pervious timestep’s Tailwater Elevation nor the previous timestep’s Tailwater Base Value are known, the Tailwater Elevation is set equal to TWBaseValueTemp.

4. If TWBaseValueTemp is not known but the previous timestep’s Tailwater Elevation is known, the current timestep’s Tailwater Elevation is set equal to the previous timestep’s Tailwater Elevation.

5. If neither TWBaseValueTemp nor the previous timestep’s Tailwater Elevation are known, but the previous timestep’s Tailwater Base Value is known, the current timestep’s Tailwater Elevation is set equal to the previous timestep’s Tailwater Base Value.

6. If the previous timestep is the initial timestep of the run, and none of the following are known: TWBaseValueTemp, the previous timestep’s Tailwater Base Value, and the previous timestep’s Tailwater Elevation, a **RiverWare**™ error is posted and the simulation is aborted.

7. If the previous timestep is NOT the initial timestep of the run and none of the following are known: TWBaseValueTemp, the previous timestep’s Tailwater Base Value, and the previous timestep’s Tailwater Elevation, the method is exited and waits for more information.

Definitions of some of the terms used in the above equations are located below:

- **TWBaseValueTemp**: a temporary value for the Tailwater Base Value. This value may be determined from another method or propagated across a link.

- **tempflow**: a local variable used as the outflow from the Reservoir.
- **TWCompare**: a local variable used to hold a value that is compared to the Tailwater Elevation obtained by table interpolation.
- **Tailwater Elevation (-1)**: the Tailwater Elevation at the previous timestep.
- **Tailwater Base Value (-1)**: the Tailwater Base Value at the previous timestep.

23.1.7.7 Coefficients Table

The Coefficients Table method multiplies Outflow, Tailwater Base Value, and Tailwater Elevation at the current and/or previous timestep by coefficients that are stored on a table. These products are added together along with constants to compute the Tailwater Elevation.

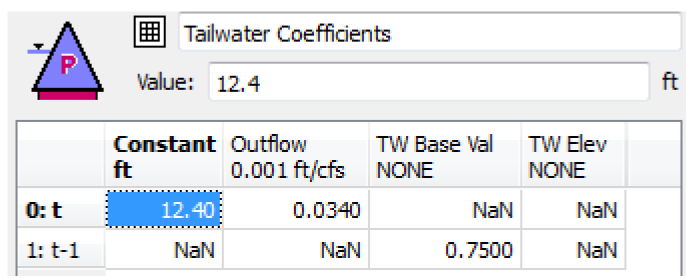
Slots Specific to this Method

☛ Tailwater Base Value

Type: Series
Units: Length
Description: base elevation of the tailwater
Information:
I/O: Optional; can be input or set by a link.
Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

☛ Tailwater Coefficients

Type: Table
Units: Length, Length/Flow, None, None
Description: This table contains the coefficients used in the calculation. The columns are as follows: **Constant**, **Outflow**, **TW Base Val**, **TW Elev**. The first row (**t**) represents the value to multiply by the current timestep's value. The second row (**t-1**)represents the value to multiply by the previous timestep's value.



	Constant ft	Outflow 0.001 ft/cfs	TW Base Val NONE	TW Elev NONE
0: t	12.40	0.0340	NaN	NaN
1: t-1	NaN	NaN	0.7500	NaN

Information: Not every cell will have a value, but there must be at least one value in the table. In addition, the coefficient for **TW Elev** at t cannot be non-zero or an error will be issued. (Tailwater Elevation at the current timestep is the value being computed, so it is not possible to use it in the calculation). If any

coefficents are specified (non-zero) for t-1 for **Outflow**, **TW Base Val**, or **TW Elev**, then the initial timestep value of those slots must also be specified.
I/O: Input only
Links: Not Linkable

At the beginning of the run, the **Tailwater Coefficients** table is verified for valid coefficients as described in the slot information above. Not every coefficient has to be specified.

When the tailwater method is executed, the following computation is performed.

$$\begin{aligned}\text{Tailwater Elevation} \;=\; & \text{Constant Coeff}[t] + \text{Constant Coeff}[t-1] + \\ & \text{Outflow Coeff}[t] \times \text{flow} + \text{Outflow Coeff}[t-1] \times \text{Outflow}[t-1] + \\ & \text{TW Base Val Coeff}[t] \times \text{TailwaterBaseValueTemp}[t] + \\ & \text{TW Base Val Coeff}[t-1] \times \text{Tailwater BaseValue}[t-1] + \\ & \text{TW Elev Coeff}[t-1] \times \text{Tailwater Elevation}[t-1]\end{aligned}$$

Each coefficient and each series value are checked for validity before including in the total.

In the above equation, the values for **flow** and **TailwaterBaseValueTemp** are based on the information that is known:

- The *flow* variable represents the outflow passed into the method if valid, or the value on the Outflow slot if not passed in. If neither is valid, Tailwater Elevation is set to the previous Tailwater Elevation. If that is not valid, the method exits and waits for a flow value to become valid.

The *TailwaterBaseValueTemp* is either the base value passed into the method or the value on the Tailwater Base Value slot. If neither is known, the previous Tailwater Base Value is used. If that is not valid, the method executes to wait for more information.

23.1.7.8 Hoover Tailwater

The Hoover Tailwater method was developed by the USBR for use on Lake Mead. This method uses a table of empirically derived constants.

Slots Specific to this Method

☛ **Tailwater Base Value**
Type: SeriesSlot
Units: LENGTH
Description: base elevation of the tailwater
Information:
I/O: Optional; can be input or set by a link.
Links: May be linked to either the Pool Elevation slot or the Backwater Elevation slot of a downstream Reservoir.

☛ **Tailwater Table**

Type: Table
Units: FLOW VS. LENGTH
Description: Reservoir Outflow vs. either the Tailwater Elevation or the tailwater elevation increment
Information: If the Tailwater Base Value is non-zero, the Tailwater Table holds values of incremental increase in Tailwater Elevation over the Base value. Otherwise, the table holds the Tailwater Elevation values.
I/O: Required input
Links: Not linkable

☛ **Hoover Tailwater Table**

Type: Table
Units: NO UNITS
Description: constants for equation that calculates the Tailwater Elevation
Information: This method was developed by the USBR for use on Lake Mead. Column 1 is the TWaverage in feet, column 2 is hte5, column 3 is coeff1.
I/O: Required input
Links: Not linkable

The Hoover Tailwater user method performs calculations to determine the Tailwater Elevation. This method is described below:

The first step of this method is to set the temporary Tailwater Base Value (TWBaseValueTemp). TWBaseValueTemp is set equal to the Tailwater Base Value if the Tailwater Base Value was calculated by another user method or set via a link.

Then the previous timestep's Tailwater Base Value is checked. If the previous timestep's Tailwater Base Value is not known, a **RiverWare**™ error is posted and the run is aborted.

The TWaverage, a local variable, is then calculated using the following steps:

1. If TWBaseValueTemp is known,

$$\text{TWaverage} = (\text{TWBaseValueTemp} + \text{Tailwater Base Value (-1)}) / 2$$

2. If the temporary TWBaseValueTemp is NOT known, the TWaverage is set equal to the previous timestep's Tailwater Base Value.

Once the TWaverage has been calculated, it is used to interpolate the empirical constants for the following equation from the Hoover Tailwater Table.

Finally, $\text{TELEFT} = \text{hte5} + ((\text{Qcfs}) / 1000 - 5) \times \text{coeff1}$

Tailwater Elevation is then set equal to TELEFT (which is converted into internal units).

Definitions of some of the terms used in the above equations are located below:

- **TWBaseValueTemp**: a temporary value for the Tailwater Base Value. This value may be determined from another method or propagated across a link.
- **Tailwater Base Value (-1)**: the Tailwater Base Value at the previous timestep.
- **TWaverage**: a local variable that is used as the average of the Tailwater Elevation over the timestep.
- **TELEFT**: the Tailwater Elevation in feet as determined by the equation given above.
- **hte5**: an empirical constant developed by the USBR.
- **Qcfs**: the flow in cfs.
- **coeff1**: an empirical constant developed by the USBR.

23.1.8 Spill

The Spill methods (except the Monthly Spill which is described in Section 23.1.8.2) calculate the Spill from Reservoirs based on several possible physical combinations of controlled and uncontrolled spillways.

The **Regulated Spill** and **Bypass** slots are regulated (i.e. controlled) spill structures. Values in these two slots can be specified by the user via inputs or rules. Each slot accommodates spill up to the maximum amount as specified by its rating table (**Regulated Spill Table** and **Bypass Table**). **Unregulated Spill** is an uncontrolled spill. Therefore, it is always a computed output based on the average **Pool Elevation** of the reservoir as specified in the **Unregulated Spill Table**. Thus, the user selects a **Spill** method based on the combination of structures (Regulated Spill, Bypass, and/or Unregulated Spill) that exist on the reservoir and the level of granularity desired.

The total **Spill** slot is the sum of the individual spills from each structure. Spills are calculated twice for each timestep. The first time a **Spill** method is called from a dispatch method, it checks for user inputs, calculates any **Unregulated Spill**, and sets the spill to zero for **Regulated Spill** and **Bypass** structures where there is no user-specified value. The total **Spill** is then calculated and returned to the dispatch method. The dispatch method determines (**Turbine**) **Release** by subtracting the **Spill** from **Outflow**, and executes the user-specified power calculation method. (On the power reservoirs, the slot is called **Turbine Release**, on the storage reservoir, the slot is called **Release**. In this description, we use the term (**Turbine**) **Release**.) If the (**Turbine**) **Release** cannot be met in the power calculation method, a second call is made to the spill calculation method. The excess flow is then distributed among the **Regulated Spill** and/or **Bypass** structures which have available capacity. If both **Regulated Spill** and **Bypass** are available, excess spill is typically first discharged through the **Regulated Spill** (except when the "**Bypass, Regulated and Unregulated**" method is selected; for this method the **Bypass** gets spill first).

The optional DRIFT flag is available on the **Regulated Spill** and **Bypass** slots. When the DRIFT flag is set for several sequential timesteps, the method models varying flow through a set spillway gate in response to fluctuations in **Pool Elevation**. The first timestep prior to initializing drift is used to determine a gate index called **Regulated** (or **Bypass**) **Drift Index**. This index is interpolated from the 3-dimensional **Regulated** (or **Bypass**) **Spill Index Table**, which relates **Pool Elevation** to **Spill** for several gate indices. In the subsequent timesteps where the DRIFT flag is set, the same index is used to find the spill value at the current average **Pool Elevation**. The gate index is maintained throughout the selected time period. At each timestep, a new value of spill is calculated for the structure based on the current **Pool Elevation**. Specifying DRIFT is considered an input, and may affect over determination of spill parameters.

23.1.8.1 None

None is used if **Spill** should not be modeled. In this method, the **Spill** slot on the reservoir is set equal to zero. All releases must be through the (**Turbine**) **Release**.

Slots Specific to this Method

☛ None

This method sets the **Spill** to zero and performs no further calculations. If the method is being called for the second time in a timestep because there is excess outflow that won't fit through (**Turbine**) **Release**, an error will be posted which states, "No Spillways Available." In this case, either decrease the **Outflow** or select a different spill method.

23.1.8.2 Monthly Spill

The Monthly Spill method is only appropriate for use in long timestep models where Reservoir fluctuations over the timestep cannot be accurately determined. It is important to note that there is no physical (head dependent) basis to the spill in this method. In this method, there are three components to the spill: unregulated, regulated and bypass. Both Regulated Spill and Bypass are considered controlled releases. A Maximum Controlled Release must be specified by the user.

$$Maximum\ Controlled\ Release = max\ Turbine\ Release + max\ Regulated\ Spill + Bypass$$

Any additional Outflow is immediately categorized as Unregulated Spill.

$$Unregulated\ Spill = Max(0.0, Outflow - Maximum\ Controlled\ Release)$$

Bypass may be specified as a user input. If not input, **RiverWare**™ sets this slot to zero. Regulated Spill is always computed by **RiverWare**™. It is set to zero unless the Reservoir cannot release the Outflow through the Release and Bypass. When this occurs, the additional portion of the Outflow is released through the Regulated Spill.

Slots Specific to this Method

☛ Bypass

Type: Series Slot
Units: FLOW
Description: flow through the Bypass spillway
Information:
I/O: Optional; may be input by the user or set to zero by **RiverWare**™.
Links: Usually not linked

☛ Maximum Controlled Release

Type: Table Slot
Units: FLOW
Description: the maximum amount of Turbine Flow, Regulated Spill, and Bypass
Information: 1X1 table slot
I/O: Required input
Links: Not linkable

☛ Regulated Spill

Type: Series Slot
Units: FLOW
Description: excess Outflow not released through the turbine(s)
Information:
I/O: Output only
Links: Usually not linked

☛ Unregulated Spill

Type: Series Slot
Units: FLOW
Description: Outflow in excess of the Maximum Controlled Release
Information:
I/O: Output only
Links: Not linkable

Initially, the Monthly Spill method is called before Release has been calculated. If the Outflow is greater than the Maximum Controlled Release, Unregulated Spill is set as:

$$Unregulated\ Spill = Outflow - Maximum\ Controlled\ Release$$

If the Outflow is less than the Maximum Controlled Release, Unregulated Spill is set to zero. In both cases, all of the following evaluations are also made:

$$Regulated\ Spill = 0.0$$

$$Bypass = user\ input\ or\ 0.0$$

$$Spill = Unregulated\ Spill + Regulated\ Spill + Bypass$$

After Release is calculated, the Monthly Spill method may be called a second time. The method is called a second time if the Release cannot accommodate the remaining portion of the Outflow:

$$Outflow > Unregulated\ Spill + Release + Bypass$$

Remember that Unregulated Spill was calculated before the Release was calculated.

If this occurs, Regulated Spill and Spill are reevaluated as follows:

$$Regulated\ Spill = Outflow - Unregulated\ Spill - Release - Bypass$$

$$Spill = Unregulated\ Spill + Regulated\ Spill + Bypass$$

23.1.8.3 Unregulated

The Unregulated spill method models a single uncontrolled spillway called **Unregulated Spill**. The **Unregulated Spill** is a function of the average reservoir **Pool Elevation**. Because it is uncontrolled, it takes precedence (i.e. water goes through it first) over other types of outflow (i.e. **Release** or **Turbine Release**) in the reservoir. When this method is chosen the user category Unregulated Spill Type , Section 23.1.9, will appear.

The user may not specify (input or via rules) any spill slots with this method.

SLOTS SPECIFIC TO THIS METHOD

☛ UNREGULATED SPILL

Type: Series Slot
Units: FLOW
Description: spill corresponding to the average Pool Elevation over the timestep
Information:
I/O: Output only
Links: Not linkable

☛ UNREGULATED SPILL CAPACITY FRACTION

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Unregulated Spill structure that is available.
Information: If not input or set by a rule, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 50 ft of a 1000 ft long crest is blocked, the Unregulated Spill Capacity Fraction would be input to 0.95.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ UNREGULATED SPILL TABLE

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation versus corresponding Unregulated Spill values
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required input

Links: Not linkable

When the Unregulated spill method is called for the first time from the Dispatch Method, Unregulated Spill is calculated and Spill is set equal to Unregulated Spill.

THE STEPS FOR COMPUTING UNREGULATED SPILL ARE GIVEN BELOW:

1. A temporary variable called "initHW" is created to represent the Pool Elevation at the beginning of the timestep. Likewise, "endHW" is created to represent the Pool Elevation at the end of the timestep (If the Pool Elevation at the end of the timestep is not known, endHW is set equal to the Pool Elevation at the beginning of the timestep.)

2. The Unregulated Spillway Crest is set equal to the Pool Elevation that corresponds to a Spill of zero (from the Unregulated Spill Table).

3. If both initHW and endHW are less than or equal to the Unregulated Spillway Crest, Unregulated Spill is set equal to zero.

4. If both initHW and endHW are greater than or equal the Unregulated Spillway Crest, the average Pool Elevation is used to determine the Unregulated Spill from the Unregulated Spill Table.

$$\text{Unregulated Spill} = \text{Value from table} \times \text{Unregulated Spill Capacity Fraction}$$

5. If either initHW or endHW is greater than the Unregulated Spillway Crest and the other is lower than the crest, the following evaluations and computations are performed:

$$\text{maxHW} = \text{the greater of initHW and endHW}$$

$$\text{minHW} = \text{the lesser of initHW and endHW}$$

$$\text{avgHW} = \frac{\text{maxHW} + \text{Unregulated Spillway Crest}}{2}$$

$$\text{spill fraction} = \frac{\text{maxHW} - \text{Unregulated Spillway Crest}}{\text{maxHW} - \text{minHW}}$$

That is, spill fraction corresponds to the fraction of the timestep during which spill occurs.

A temporary variable called "temp spill" is obtained from the linear interpolation of the Unregulated Spill Table using avgHW. Unregulated Spill is then calculated as:

$$\text{Unregulated Spill} = \text{spill fraction} \times \text{temp spill} \times \text{Unregulated Spill Capacity Fraction}$$

The Unregulated Spill is then limited to be less than or equal to the maxUnregulatedSpill if one has been calculated. See below.

The Unregulated spill method will be called a second time (after the (**Turbine**) **Release** has been calculated) only if the sum of (**Turbine**) **Release** and **Spill** are less than **Outflow**. When this is the case, an error which reads, "Outflow greater than spillway capacities and Release" is posted because the excess **Spill** cannot be incorporated.

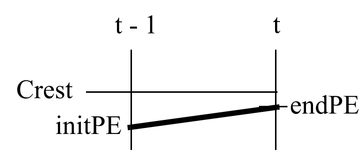
Computing the Maximum Possible Unregulated Spill

If an unregulated spill method is called from the solveMB_givenInflowRelease dispatch method, getMinSpillGivenInflowRelease or solveTurbineRelGivenEnergyInflow RPL predefined functions, there is an upper limit on the unregulated spill. The upper limit prevents the unregulated spill from dropping the reservoir below the crest. The following algorithm is performed at the start of the following methods to compute this maximum unregulated spill (maxUnregulatedSpill):

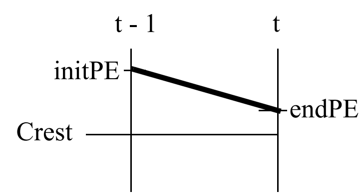
- solveMB_givenInflowRelease dispatch method (Storage and Level Power reservoirs)
- solveMB_givenInflowOutflow dispatch method (Storage and Level Power reservoirs)
- getMinSpillGivenInflowRelease RPL predefined function
- solveTurbineRelGivenEnergyInflow RPL predefined function

1. Given previous Storage and Pool Elevation (initPE), all known inflows (Inflow, Hydrologic Inflow, Precipitation, Return Flow, etc) and known outflows (Release, Regulated Spill, Bypass, Evaporation, Diversion, etc) are used to compute the storage and pool elevation (endPE) that would occur with no additional unregulated spill.

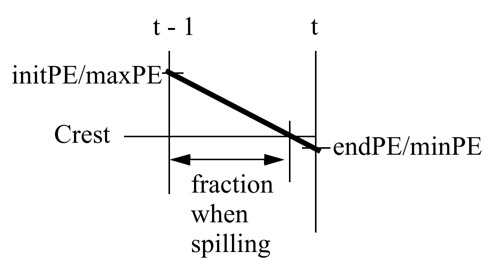
2. If both initPE and endPE are less than or equal to the Unregulated Spillway Crest, maxUnregulatedSpill is set to zero and the computation exits. There is no way there could be unregulated spill. See figure to the right.



3. If both initPE and endPE are greater than or equal to the Unregulated Spillway Crest, the maxUnregulatedSpill is computed as the flow that would draw the reservoir down to exactly reach the crest at the end of the timestep. This computation solves the reservoir mass balance and includes all source and sink terms. All water above the crest could be spilled. See figure to the right.



4. If either initPE or endPE is greater than the Unregulated Spillway Crest and the other is lower than the crest (because of existing Diversions, Evap, etc), the following evaluations and computations are performed: (See figure to the right)



maxPE = the greater of initPE and endPE

minPE = the lesser of initPE and endPE

$$\text{spill fraction} = \frac{\text{maxPE} - \text{Unregulated Spillway Crest}}{\text{maxPE} - \text{minPE}}$$

That is, spill fraction corresponds to the fraction of the timestep during which spill occurs.

5. The storage at the crest, crestStorage, is computed from the Elevation Volume Table.

6. The storage, maxStorage, that corresponds to the maxPE is found on the Elevation Volume Table.

7. The maxUnregulatedSpill (limited to be greater than or equal to zero) is then computed as:

$$\text{maxUnregulatedSpill} = \frac{\text{maxStorage} - \text{crestStorage}}{\text{TimestepLength}} \times \text{spillFraction}$$

The maxUnregulatedSpill is applied as a final limit on the Unregulated Spill. Remember, the Unregulated Spill is computed as described in the start of this method using the Unregulated Spill Table and a similar spill fraction approach; it may already be less than the maxUnregulatedSpill.

23.1.8.4 Regulated

The Regulated spill method models **Spill** using one controlled spillway called **Regulated Spill**. Because the spill is controlled, the spill may be any value between zero and the maximum possible regulated spill for that pool elevation. The user may specify (input or via rules) either:

- No slots
- **Spill** or
- **Regulated Spill**

If either is specified and there is excess flow which cannot be met by the (**Turbine**) **Release**, a **RiverWare**™ error will be flagged and the simulation halted.

Slots Specific to this Method

☛ Regulated Spill

Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare**™. If Regulated Spill is input by the user and the value is less than the required spill, a **RiverWare**™ error is flagged and the simulation is halted.

Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☛ Regulated Spill Capacity Fraction

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Regulated Spill structure that is available.
Information: If not input or set by a rule, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Regulated Spill Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index
Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Regulated Spill Index Table.
Links: Not linkable

☛ Regulated Spill Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding Maximum Regulated Spill values
Information:
I/O: Required input
Links: Not linkable

☛ Regulated Spill Index Table

Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Regulated Spill
Information: Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column

3. The table shown below is an example of the proper way to formulate the Regulated Spill Index Table.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130
4	550	180
4	600	230

I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.

Links: Not linkable

Note: Regulated Spill and Spill are both outputs if neither is specified by the user. Only one of these slots, however, can be specified on a given timestep. When this is the case, the other slot will be output.

The first step in the Regulated spill method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 23.1.10.2. If not, the minRegSpill is zero. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

Release is then checked to see if it has been calculated. If Release is not known, it means that the method is being called for the first time for the particular timestep and the following steps are taken:

1. If both the Spill and Regulated Spill are input (remember setting a DRIFT flag is considered an input), a **RiverWare**™ error is flagged and the run is aborted.

2. If Spill is input by the user, and it is greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted. Otherwise, Regulated Spill is set equal to Spill.

3. If Regulated Spill is input/rules by the user and the DRIFT flag is set, a function is called to perform the drift calculations. A description of the DRIFT function is given at the end of this method description. If Regulated Spill is input and greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted. Otherwise, Spill is set equal to the Regulated Spill.

4. If neither Regulated Spill nor Spill are input, they are both set equal to the minRegSpill.

After Release has been calculated, the Regulated spill function may be called a second time if the sum of Release and Spill is less than the Outflow.

The following calculations and evaluations are performed if the function is called for the second time:

1. If either Spill or Regulated Spill are input, a **RiverWare**™ error is posted stating that **RiverWare**™ is unable to allocate the excess flow and the run is aborted.

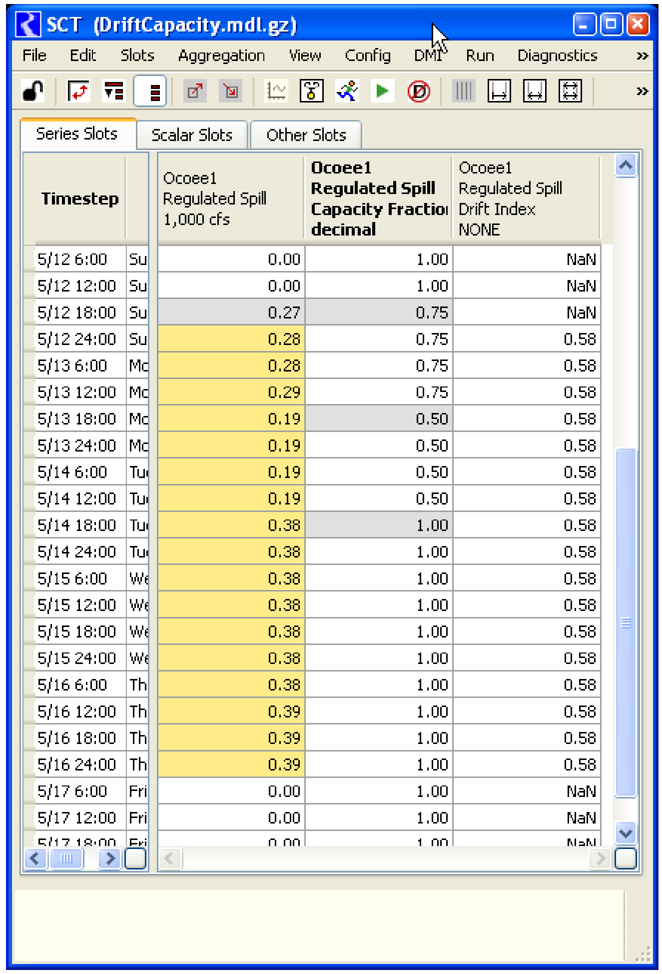
2. Regulated Spill is set equal to the Outflow minus the Release. If Regulated Spill is greater than the maxRegSpill, a **RiverWare**™ error is flagged informing the user that Outflow is greater than the spillway capacities and Release and the run is aborted.

3. Spill is set equal to Regulated Spill.

DRIFT CALCULATIONS:

The drift function is used to calculate Regulated Spill at a specific timestep if it is flagged DRIFT. If the current timestep's Regulated Drift Index is not known, but the previous timestep's Regulated Drift Index is known, the current Regulated Drift Index is set equal to the previous timestep's Regulated Drift Index.

The Drift tables assume that the full spill works is available. Therefore, if there is a Capacity Fraction that is less than 1.0, the Drift calculation (for both regulated and bypass) must be modified.



Timestep		Ocoee1 Regulated Spill 1,000 cfs	Ocoee1 Regulated Spill Capacity Fractio decimal	Ocoee1 Regulated Spill Drift Index NONE
5/12 6:00	Su	0.00	1.00	NaN
5/12 12:00	Su	0.00	1.00	NaN
5/12 18:00	Su	0.27	0.75	NaN
5/12 24:00	Su	0.28	0.75	0.58
5/13 6:00	Mc	0.28	0.75	0.58
5/13 12:00	Mc	0.29	0.75	0.58
5/13 18:00	Mc	0.19	0.50	0.58
5/13 24:00	Mc	0.19	0.50	0.58
5/14 6:00	Tu	0.19	0.50	0.58
5/14 12:00	Tu	0.19	0.50	0.58
5/14 18:00	Tu	0.38	1.00	0.58
5/14 24:00	Tu	0.38	1.00	0.58
5/15 6:00	We	0.38	1.00	0.58
5/15 12:00	We	0.38	1.00	0.58
5/15 18:00	We	0.38	1.00	0.58
5/15 24:00	We	0.38	1.00	0.58
5/16 6:00	Th	0.38	1.00	0.58
5/16 12:00	Th	0.39	1.00	0.58
5/16 18:00	Th	0.39	1.00	0.58
5/16 24:00	Th	0.39	1.00	0.58
5/17 6:00	Fri	0.00	1.00	NaN
5/17 12:00	Fri	0.00	1.00	NaN
5/17 18:00	Fri	0.00	1.00	NaN

With the Drift flag is set, if there is a valid Capacity Fraction[t-1] that is not equal to 1.0, then the Capacity Fraction[t] is set to the previous value, but not overwriting inputs or rule values. This causes the Capacity Fraction to remain throughout the drift operation unless it is changed via a new user input. The screenshot to the right shows a sample run. The Capacity fraction is set to 0.75 on 5/12 18:00 and that value remains until a new value is set via user input on 5/13 18:00. Although this set of inputs may not make physical or operations sense, is shows how the algorithm would perform given the inputs shown.

The current Regulated Drift Index is then used in conjunction with the average Pool Elevation over the current timestep and the Regulated Spill Index Table to obtain the current timestep's Regulated Spill.

If it is the first DRIFT timestep, a Gate Index must be calculated. This is done by using an average of the Pool Elevation at t-2 (when available) and at t-1, the previous regulated spill, the greatest and least spill values, and the Regulated Spill Index Table. If there is a non-zero capacity fraction, the value used for previous regulated spill is adjusted as follows:

$$\text{Regulated Spill}[t-1]_{Adj} = \frac{\text{Regulated Spill}[t-1] - \text{Closed Gate Overflow}[t-1]}{\text{Regulated Spill Capacity Fraction}[t-1]}$$

This computes the spill that would have occurred if all the gates were available. If the previous regulated spill is less than the smallest possible Regulated Spill or greater than the largest possible Regulated Spill (according to the Regulated Spill Index Table), a **RiverWare**™ error is flagged and the run is aborted.

The Regulated Drift Index slot is then set for the current timestep. Finally, the Regulated Spill can be determined from the three dimensional interpolation of the Regulated Spill Index Table using the average Pool Elevation over the timestep and the Regulated Spill Index.

At the end of the algorithm, the computed regulated spill (assuming all the gates are available) is multiplied by the Capacity Fraction [t] to determine spill that will occur with the given capacity fraction. This Regulated Spill is then set on the slot.

23.1.8.5 Regulated and Unregulated

The Regulated and Unregulated method models **Spill** through one controlled, **Regulated Spill**, and one uncontrolled spillway, **Unregulated Spill**. First, the **Unregulated Spill** can not be specified (input or via rules) or a **RiverWare**™ error will abort the run. The **Unregulated Spill** is a function of the average reservoir **Pool Elevation** and takes precedence (i.e. water goes through it first) over other types of outflow (i.e. **Release** or **Turbine Release** or **Regulated Spill**) in the reservoir. When this method is chosen the user category Unregulated Spill Type , Section 23.1.9, will appear.

Second, the user may specify (input or via rules) either

- No slots
- **Spill** or
- **Regulated Spill**

If one is specified and there is excess flow which cannot be met by the (**Turbine**) **Release**, an error will be flagged and the simulation halted.

SLOTS SPECIFIC TO THIS METHOD

☛ REGULATED SPILL

Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare**™. If regulated spill is set by the user and the value is greater than the required spill, an error is flagged and the simulation is halted.
Links: Usually not linked. But, it can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☛ REGULATED SPILL CAPACITY FRACTION

Type: Series Slot
Units: DECIMAL

Description: The fraction of the Regulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Regulated Spill Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index
Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Regulated Spill Index Table.
Links: Not linkable

☛ Regulated Spill Index Table

Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Regulated Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.
Links: Not linkable

Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column 3.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130

Gate Index	Pool Elevation	Spill
4	550	180
4	600	230

☛ REGULATED SPILL TABLE

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding Max Regulated Spill values
Information:
I/O: Required input
Links: Not linkable

☛ UNREGULATED SPILL

Type: Series Slot
Units: FLOW
Description: spill corresponding to the average Pool Elevation over the timestep
Information:
I/O: Output only
Links: Not linkable

☛ UNREGULATED SPILL CAPACITY FRACTION

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Unregulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 50 ft of a 1000 ft long crest is blocked, the Unregulated Spill Capacity Fraction would be input to 0.95.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ UNREGULATED SPILL TABLE

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding Unregulated Spill values
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required input
Links: Not linkable

Note: Regulated Spill and Spill are both output slots if neither is input by the user. Only one of these slots, however, can be input for any timestep. When this is the case, the other slot will be an output slot.

The first step in the regulatedPlusUnregSpillCalc method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 23.1.10.2. If not, the minRegSpill is 0.0. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

The Unregulated Spill is then calculated through the steps described in the “Unregulated”. If the Unregulated Spill is input by the user, a **RiverWare**™ error is flagged and the run is aborted.

Release is then checked to see if it has been calculated. If Release is not known, it means that the method is being called for the first time for the particular timestep and the following steps are taken:

1. If both the Spill and Regulated Spill are input (remember setting a DRIFT flag is considered an input), a **RiverWare**™ error is flagged and the run is aborted.

2. If Spill is input and it is greater than the sum of the Unregulated Spill and the maxRegSpill, a **RiverWare**™ error is flagged which states that the requested Spill cannot be met. If the Spill is input and less than the Unregulated Spill plus minRegSpill, a **RiverWare**™ error is flagged. Otherwise, Regulated Spill is calculated as Spill minus Unregulated Spill.

3. If Regulated Spill is input by the user and the DRIFT flag is set, a function is called to perform the drift calculations. A description of the DRIFT function is given in the Regulated Spill section. If Regulated Spill is input and greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted. Otherwise, Spill is set equal to the Regulated Spill plus Unregulated Spill.

4. If neither Regulated Spill nor Spill are input, Regulated Spill is set equal to the minRegSpill and Spill is set equal to Unregulated Spill plus minRegSpill.

After the Release has been calculated, the Regulated and Unregulated function may be called a second time if the sum of Release and Spill is less than the Outflow.

The following calculations are performed if the function is called for the second time:

1. If either Spill or Unregulated Spill are input, an error is posted because there are no free spill variables and **RiverWare**™ is unable to allocate the excess Outflow.

2. The Regulated Spill is calculated using the following equation:

$$\text{Regulated Spill} = \text{Outflow} - \text{Turbine Release} - \text{Unregulated Spill}$$

3. If Regulated Spill is greater than the maxRegSpill, a **RiverWare™** error is posted stating that the Outflow is greater than the spillway capacities and Release and the run is aborted.

4. Spill is calculated as Regulated Spill plus Unregulated Spill.

23.1.8.6 Regulated and Bypass

The Regulated and Bypass method models spill through two regulated spillways called **Regulated Spill** and **Bypass**. The user may specify (input or via rules):

- No slots
- **Spill**
- **Spill** and **Bypass**
- **Spill** and **Regulated Spill**
- **Bypass**
- **Regulated Spill**, or
- **Bypass** and **Regulated Spill**

If all three slots are specified, an error will be issued. Also, if **Spill** is specified and there is excess flow that cannot be met by (**Turbine**) **Release**, a **RiverWare™** error will be flagged and the simulation halted.

The order in which water goes through the various outflow structures depends on what is known. Input/Rules values take precedence, followed by (**Turbine**) **Release**, followed by **Regulated Spill**, and finally by **Bypass**. For example, on a timestep where there is zero (**Turbine**) **Release** and no spill slots are specified, outflows will first go through **Regulated Spill** and any flow greater than max regulated spill will go through **Bypass**.

Slots Specific to this Method

☛ Bypass

Type: Series Slot
Units: FLOW
Description: flow through the Bypass spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare™**. If Bypass is set by the user and the value is greater than the required spill, a **RiverWare™** error is flagged and the simulation is halted.
Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Bypass behaves the same as if it were input.

☛ Bypass Capacity Fraction

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Bypass structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 2 of 8 gates are unavailable, the Bypass Capacity Fraction would be input to 0.75.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Bypass Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index for the Bypass spillway
Information: If the user has set the DRIFT flag on the Bypass slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Bypass Index Table.
Links: Not linkable

☛ Bypass Index Table

Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Bypass Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Bypass spill slot, this data table must be provided.
Links: Not linkable

☛ Bypass Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding maximum bypass spill values
Information:
I/O: Required input
Links: Not linkable

☛ Regulated Spill

Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway

Information:

I/O: Optional; may be input by the user or determined by **RiverWare™**.

Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☛ Regulated Spill Capacity Fraction

Type: Series Slot

Units: DECIMAL

Description: The fraction of the Regulated Spill structure that is available.

Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.

I/O: Input, set by a rule, or output

Links: Not linkable

☛ Regulated Spill Drift Index

Type: Series Slot

Units: NONE

Description: gate setting index for the Regulated Spill

Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.

I/O: Optional; if not set by the user, the index is calculated from the Regulated Spill Index Table.

Links: Not linkable

☛ Regulated Spill Index Table

Type: Table Slot

Units: NOUNITS vs. LENGTH vs. FLOW

Description: Gate Index vs. Pool Elevation vs. Regulated Spill

Information:

I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.

Links: Not linkable

Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column 3.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130
4	550	180
4	600	230

☛ Regulated Spill Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding maximum regulated spill values
Information:
I/O: Required input
Links: Not linkable

Note: Spill, Regulated Spill, and Bypass can all be outputs if they are not specified by the user. The user may specify Spill, or either Regulated Spill or Bypass, or both Regulated Spill and Bypass. The slots which are not specified will be output slots.

The first step in the Regulated Spill method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 23.1.10.2. If not, the minRegSpill is 0.0. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

The next step is to obtain the maximum Bypass, maxBypass, by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Bypass Table. Then the Bypass Capacity Fraction is applied as follows:

$$\text{maxBypass} = \text{value from Bypass Table} \times \text{Bypass Capacity Fraction}$$

1. If Spill, Bypass, and Regulated Spill are all input/rules, a **RiverWare**™ Error is posted and the simulation run is aborted.

2. If Spill and Regulated Spill are input/rules by the user, the following steps are performed (Remember: If the DRIFT flag is set on the Regulated Spill slot, Regulated Spill is considered an input.):
   - Drift calculations are performed if the DRIFT flag is set on the Regulated Spill slot. See Regulated Spill for a description of the drift calculations.
   - If Regulated Spill is greater than either Spill or maxRegSpill a **RiverWare**™ error is flagged and the run is aborted.
   - If Regulated Spill is less than minRegSpill, a **RiverWare**™ error is flagged and the run is aborted.
   - Bypass is calculated as Spill minus Regulated Spill.
   - If Bypass is greater than the maxBypass, an error is flagged.

3. If Spill and Bypass are input, the following steps are taken (Remember: If the DRIFT flag is set on the Bypass slot, Bypass is considered an input.):
   - Drift calculations are performed if the DRIFT flag is set on the Bypass slot. The DRIFT calculations are performed in a similar manner to the drift calculations for Regulated Spill which are explained in the Regulated Spill method description.
   - If Bypass is greater than either Spill or the maxBypass, a **RiverWare**™ error is flagged and the simulation is aborted.
   - Regulated Spill is calculated as Spill minus Bypass.
   - If Regulated Spill is either greater than the maximum regulated spill or less than the minimum regulated spill, an error is flagged and the run is aborted.

4. If Spill is input but neither Regulated Spill nor Bypass are input or flagged as DRIFT, the following steps are taken:
   - Regulated Spill is set as the lesser value of either the Spill or the maxRegspill. The Regulated Spill also cannot be less than the minRegSpill.
   - If Regulated Spill is less than Spill, Bypass is calculated as Spill minus Regulated Spill.
   - If Bypass is greater than the maxBypass, a **RiverWare**™ error is flagged and the run is aborted.
   - If Regulated Spill is equal to Spill, Bypass is set equal zero.

5. If Spill is not input, and both Bypass and Regulated Spill are input, the following steps are taken:
   - The drift calculations are performed for both Regulated Spill and Bypass if the DRIFT flags have been set. A description of the DRIFT calculations is contained in the Regulated Spill method.
   - Regulated Spill and Bypass are checked against the maxRegSpill and maxBypass, respectively. If either Regulated Spill is greater, a **RiverWare**™ error is flagged and the run is aborted.
   - Spill is calculated as Regulated Spill plus Bypass.
6. If the DRIFT flag is set on the Regulated Spill, the drift calculations are performed (as described in the Regulated Spill method) to calculate Regulated Spill. The calculated Regulated Spill value is then checked against spillway capacities. If only Regulated Spill is input, the value is checked against the spillway capacity. Spill is set equal to Regulated Spill and Bypass is set to zero if the Regulated Spill is less than or equal to the maxRegSpill. If the Regulated Spill is greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted and the simulation run is aborted.
7. If the DRIFT flag is set on the Bypass, the drift calculations are performed (as described in the Regulated Spill method) to calculate Bypass. The calculated Bypass value is then checked against spillway capacities. If only Bypass is input, the input value is checked against the spillway capacity. Spill is calculated as Bypass plus Regulated Spill. Regulated Spill is set to the minimum regulated spill, if the Bypass is less than the maximum bypass. If the Bypass is greater than the maxBypass, a **RiverWare**™ error is posted and the simulation run is aborted.
8. If no slots are input, Spill is set equal minRegSpill. Bypass is set to zero and Regulated Spill is set to minRegSpill.

After Release has been calculated, the Regulated and Bypass function may be called a second time if the sum of the Release and Spill is less than the Outflow.

The following calculations are performed if the function is called for the second time:

1. If either Spill is input, or both Regulated Spill and Bypass are input, a **RiverWare**™ error is flagged and the simulation run is aborted because there are no free spill variables.
2. If only Regulated Spill is input or flagged as DRIFT, Bypass is recalculated using the following formula:

$$\text{Bypass} = \text{Outflow} - \text{Regulated Spill} - \text{Turbine Release}$$

Bypass is then checked against its spillway capacity and a **RiverWare™** error is flagged and the simulation is aborted the spillway capacity is exceeded. Spill is calculated as Bypass plus Regulated Spill if the Bypass is less than or equal to maximum allowable bypass.

3. If only Bypass is input or flagged as DRIFT, Regulated Spill is recalculated using the following formula:

$$\text{Regulated Spill} = \text{Outflow} - \text{Bypass} - \text{Turbine Release}$$

The Regulated Spill is then checked against its spillway capacity and minRegSpill. A **RiverWare™** error is posted and the simulation run is aborted if the spillway capacity is exceeded or is less than minimum. Spill is calculated as the sum of Bypass and Regulated Spill.

4. If neither Bypass nor Regulated Spill is input, the following steps are performed:

- A local variable, excess, is calculated as Outflow minus Turbine Release minus minRegSpill.
- Regulated Spill is set equal to the lesser value of excess or maxRegSpill but must be greater than minRegSpill.
- If Regulated Spill is less than the excess, Bypass is calculated as excess minus Regulated Spill.
- Bypass is checked against its spillway capacity. If Bypass is greater than the maxBypass, a **RiverWare™** error is posted and the simulation run is aborted.
- Spill is calculated as the Bypass plus the Regulated Spill.
- If Regulated Spill is equal to the Excess, Bypass is set equal to zero and Spill is set equal to Regulated Spill.

23.1.8.7 Regulated, Bypass and Unregulated

This method models spill through two controlled spillways called **Bypass** and **Regulated Spill** and one uncontrolled spillway called **Unregulated Spill**. The user may not specify (input or via rules) the **Unregulated Spill**. This value is always output and is a function of the average reservoir Pool Elevation. The user may specify (input or rules):

- No slots
- **Spill**
- **Spill** and **Bypass**
- **Spill** and **Regulated Spill**
- **Bypass**
- **Regulated Spill**, or
- **Bypass** and **Regulated Spill**

If **Spill**, **Regulated Spill**, and **Bypass** are specified, an error will be issued. Also, if **Spill** is specified and there is excess flow that cannot be met by (**Turbine**) **Release**, a **RiverWare**™ error will be flagged and the simulation halted.

The order in which water goes through the various outflow structures depends on what is known. **Unregulated Spill** takes precedence, followed by input/rules values, followed by (**Turbine**) **Release**, followed by **Regulated Spill**, and finally by **Bypass**. For example, on a timestep where there is zero (**Turbine**) **Release** and no spill slots are specified, outflows will first go through **Unregulated Spill** (computed based on pool elevation), then **Regulated Spill** up to capacity and any excess flows will go through **Bypass**.

Slots Specific to this Method

☛ Bypass

Type: Series Slot
Units: FLOW
Description: flow through the Bypass spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare**™. If Bypass is set by the user and the value is greater than the required spill, a **RiverWare**™ error is flagged and the simulation is halted.
Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Bypass behaves the same as if it were input.

☛ Bypass Capacity Fraction

Type: Series Slot
Units: Decimal
Description: The fraction of the Bypass structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 2 of 8 gates are unavailable, the Bypass Capacity Fraction would be input to 0.75.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Bypass Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index for the Bypass spillway
Information: If the user has set the DRIFT flag on the Bypass slot, the gate setting index from the previous timestep is maintained.

I/O: Optional; if not set by the user, the index is calculated from the Bypass Index Table.

Links: Not linkable

☛ Bypass Index Table

Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Bypass Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Bypass spill slot, this data table must be provided.

Links: Not linkable

☛ Bypass Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding maximum bypass spill values
Information:
I/O: Required input

Links: Not linkable

☛ Regulated Spill

Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare**™.

Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☛ Regulated Spill Capacity Fraction

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Regulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.
I/O: Input, set by a rule, or output

Links: Not linkable

☛ Regulated Spill Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index for the Regulated Spill
Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if this slot is not set by the user, the gate index is calculated from the Regulated Spill Index Table.
Links: Not linkable

☛ Regulated Spill Index Table

Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Regulated Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.
Links: Not linkable

Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column 3.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130
4	550	180
4	600	230

☛ Regulated Spill Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding maximum regulated spill values
Information:

I/O: Required input
Links: Not linkable

☛ Unregulated Spill

Type: Series Slot
Units: FLOW
Description: spill corresponding to the average Pool Elevation over the timestep
Information:
I/O: Output only
Links: Not linkable

☛ Unregulated Spill Capacity Fraction

Type: Series Slot
Units: Decimal
Description: The fraction of the Unregulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 50 ft of a 1000 ft long crest is blocked, the Unregulated Spill Capacity Fraction would be input to 0.95.
I/O: Input, set by a rule, or output
Links: Not linkable

☛ Unregulated Spill Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding unregulated spill values
Information:
I/O: Required input
Links: Not linkable

Note: Spill, Regulated Spill, and Bypass may be output slots if they are not specified as input by the user. The user may specify either Spill or Regulated Spill or Bypass, or both Unregulated Spill and Bypass as input. The slots which are not set as input will be output slots.

The first step in the Regulated Spill method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 23.1.10.2. If not, the minRegSpill is 0.0. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

The next step is to obtain the maximum Bypass, maxBypass by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Bypass Table. Then the Bypass Capacity Fraction is applied as follows:

$$\text{maxBypass} = \text{value from Bypass Table} \times \text{Bypass Capacity Fraction}$$

The Unregulated Spill is then calculated through the steps described in the "Unregulated". If the Unregulated Spill is input by the user, a **RiverWare**™ error is flagged and the run is aborted.

If Spill, Bypass, and Regulated Spill are input, Spill is overdetermined and a **RiverWare**™ error is flagged and the simulation run is aborted.

If Release has not been calculated, the method is executing for the first time in the current timestep and the following steps are taken:

1. If Spill is input by the user and Unregulated Spill is greater than the Spill, a **RiverWare**™ error is flagged and the simulation run is aborted.
2. If Spill and Regulated Spill are input the following steps are taken (Remember: If the DRIFT flag is set on the Regulated Spill slot, Regulated Spill is considered an input):
   - Drift calculations are performed if the DRIFT flag is set on the Regulated Spill slot. See Regulated Spill for a description of the drift calculations.
   - If Regulated Spill is greater than either Spill or maxRegSpill, a **RiverWare**™ error is flagged and the run is aborted.
   - If Regulated Spill is less than minRegSpill, a **RiverWare**™ error is flagged and the run is aborted.
   - Bypass is calculated as Spill minus Regulated Spill minus Unregulated Spill.
   - If Bypass is either greater than the maxBypass or less than zero, an error is flagged.
3. If Spill and Bypass are input, the following steps are taken (Remember: If the DRIFT flag is set on the Bypass slot, Bypass is considered an input.):
   - Drift calculations are performed if the DRIFT flag is set on the Bypass slot. The DRIFT calculations are performed in a similar manner to the drift calculations for Regulated Spill, which are explained in the Regulated Spill method description.
   - If Bypass is greater than either Spill or the maxBypass, a **RiverWare**™ error is flagged and the run is aborted.

- Regulated Spill is calculated as Spill minus Bypass minus Unregulated Spill.
- If Regulated Spill is either greater than the maxRegSpill or less than the minRegSpill, an error is flagged and the run is aborted.

4. If Spill is input by the user and neither Bypass nor Regulated Spill are input, the following steps are taken:

- Regulated Spill is set as the lesser value of either the Spill minus Unregulated Spill or the maxRegSpill. The Regulated Spill also cannot be less than the minRegSpill.
- If Regulated Spill is less than Spill minus Unregulated Spill, Bypass is set equal to Spill minus Regulated Spill minus Unregulated Spill.
- If Bypass is greater than the maxBypass, a **RiverWare**™ error is flagged and the run is aborted.
- If Regulated Spill is equal to Spill minus Unregulated Spill, Bypass is set equal to zero.

5. If Spill is not input, but both Bypass and Regulated Spill are input, the following steps are taken:

- The drift calculations are performed for both Regulated Spill and Bypass if the DRIFT flags have been set. A description of the DRIFT calculations is contained in the Regulated Spill method.
- Regulated Spill and Bypass are checked against the maxRegSpill and maxBypass, respectively. If either Regulated Spill is greater, a **RiverWare**™ error is flagged and the run is aborted. If Regulated Spill is less than the minRegSpill, an error is issued.
- Spill is calculated as the sum of Regulated Spill, Bypass, and Unregulated Spill.

6. If the DRIFT flag is set on the Regulated Spill, the drift calculations are performed (as described in the Regulated Spill method) to calculate Regulated Spill. The calculated Regulated Spill Value is then checked against spillway capacities. If only Regulated Spill is input, the value is checked against the spillway capacity. Spill is calculated as Regulated Spill plus Unregulated Spill and Bypass is set to zero, if the Regulated Spill is less than the maxRegSpill. If the Regulated Spill is greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted and the simulation run is aborted.

7. If the DRIFT flag is set on the Bypass, the drift calculations are performed (as discussed in the Regulated Spill method) to calculated the Bypass. The calculated Bypass value is then checked against spillway capacities. If only Bypass is input, the input value is checked against the spillway capacity. Spill is calculated as Bypass plus Unregulated Spill and Regulated Spill is set to the minRegSpill, if the Bypass is less than the maxBypass. If the Bypass is greater than the maxBypass, a **RiverWare™** error is posted and the simulation run is aborted.

8. If no slots are input, Spill is set equal to Unregulated Spill plus minRegSpill. Bypass is set to zero and Regulated Spill is set to minRegSpill.

After Release has been calculated, the Regulated, Bypass and Unregulated function may be called a second time if the sum of the Release and Spill is less than the Outflow.

The following calculations are performed if the function is called for the second time:

1. If either Spill is input or both Regulated Spill and Bypass are input, a **RiverWare™** error is flagged and the simulation run is aborted because there are no free spill variables.

2. If only Regulated Spill is input or flagged as DRIFT, Bypass is recalculated using the following formula:

$$\text{Bypass} = \text{Outflow} - \text{Regulated Spill} - \text{Unregulated Spill} - \text{Turbine Release}$$

The Bypass is then checked against its spillway capacity. A **RiverWare™** error is posted and the simulation run is aborted if the spillway capacity is exceeded. Spill is calculated as the sum of Bypass, Unregulated Spill, and Regulated Spill if the Bypass is less than or equal to the maxBypass.

3. If only Bypass is input, Regulated Spill is recalculating the following formula:

$$\text{Regulated Spill} = \text{Outflow} - \text{Bypass} - \text{Unregulated Spill} - \text{Turbine Release}$$

The Regulated Spill is then checked against its spillway capacity and minRegSpill. A **RiverWare™** error is posted and the simulation run is aborted if the spillway capacity is exceeded or is less than minimum. Spill is calculated as the sum of Bypass, Regulated Spill, and Unregulated Spill if the Regulated Spill is less than or equal to the maximum allowable regulated spill.

4. If neither Regulated Spill nor Bypass are input, the following steps are performed:
   - A local variable, excess, is calculated as Outflow minus Unregulated Spill minus Turbine Release minus minimum regulated spill.
   - Regulated Spill is set equal to the lesser value of excess or maxRegSpill but must be greater than minRegSpill.
   - If Regulated Spill is less than excess, Bypass is calculated as excess minus Regulated Spill.
   - Bypass is checked against its spillway capacity. If Bypass is greater than the maxBypass, a **RiverWare™** error is posted and the simulation run is aborted.

- Spill is set equal to the sum of Bypass, Regulated Spill, and Unregulated Spill.
- If Regulated Spill is equal to Excess, Bypass is set equal to zero and Spill is set equal to the sum of Regulated Spill and Unregulated Spill.

23.1.8.8 Bypass, Regulated and Unregulated

Note: This user method is the similar to the **Regulated, Bypass and Unregulated** method but switches the order of the **Bypass** and **Regulated Spill** outlet works. This method is preferable in institutional cases where the term "**Bypass**" is favored over the term "**Regulated Spill**". Other than the order reversal, the functionality is similar to the Regulated, Bypass and Unregulated method.

This method models spill through two controlled spillways called **Bypass** and **Regulated Spill** and one uncontrolled spillway called **Unregulated Spill**. The user may not specify (input or via rules) the **Unregulated Spill**. This value is always output and is a function of the average reservoir **Pool Elevation**. The user may specify (input or rules):

- No slots
- **Spill**
- **Spill** and **Bypass**
- **Spill** and **Regulated Spill**
- **Bypass**
- **Regulated Spill**, or
- **Bypass** and **Regulated Spill**

If **Spill**, **Regulated Spill**, and **Bypass** are specified, an error will be issued. Also, if **Spill** is specified and there is excess flow that cannot be met by (**Turbine**) **Release**, a **RiverWare**™ error will be flagged and the simulation halted.

The order in which water will go through the various outflow structures depends on what is known. **Unregulated Spill** takes precedence, followed by input/rules values, followed by (**Turbine**) **Release**, followed by **Bypass**, and finally by **Regulated Spill**. For example, on a timestep where there is zero (**Turbine**) **Release** and no spill slots are specified, outflows will first go through **Unregulated Spill** (required based on pool elevation), then **Bypass** up to capacity and any excess flows will go through **Regulated Spill**.

Please see the **regPlusBypassPlusUnregSpill** method for a description of the slots particular to this method and the algorithm of this method. The algorithm for this method is only different in that **Bypass** takes precedence over **Regulated Spill**.

23.1.9 Unregulated Spill Type

This category is only visible when a method using **Unregulated Spill** is chosen. The three Unregulated Spill Types are Bare Crest Only, Two Unregulated Flow, and Three Unregulated Flows.

23.1.9.1 Bare Crest Only

The Bare Crest Only method is the default method in the Unregulated Spill Type Category. The method assumes an unobstructed spillway where the flow over the spillway is a function of the **Unregulated Spill Table**. There are no slots specifically associated with this method.

23.1.9.2 Two Unregulated Flows

When the **Two Unregulated Flows** method is selected, flow over the spillway is a function of the **Unreg Flow 2 Spill Table.** If the pool elevation meets or exceeds the **Unreg Flow 2 Failure Elevation**, flow becomes a function of the **Unregulated Spill Table.**

Note: This method originally was called **Flashboards** but was renamed to be more general. (Flashboards are wooden boards installed in the unregulated spillway so that the reservoir may store more water than what the spillways themselves would allow.)

☛ Unreg Flow 2 Avail and Failure Time

Type: Agg Series Slot
Units: FRACTION
Description: Availability of Unreg Flow 2 Spill Table, fraction of timestep when Unreg Flow 2 Spill Table is in use.
Information:
I/O: Optional: Availability may be input by user or set by a rule. Failure time is output only
Links: Not linkable

☛ Unreg Flow 2 Failure Elevation

Type: Table Slot
Units: LENGTH
Description: Pool Elevation at which Unreg Flow 2 Spill Table is no longer used.
Information:
I/O: Required Input
Links: Not linkable

☛ Unreg Flow 2 Spill Table

Type: Table Slot

Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding unregulated flow 2 spill values.
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required Input
Links: Not linkable

METHOD DETAILS

To determine the spill tables to use during a timestep based on availability of Unreg Flow 2, the following three possibilities are checked:

1. Availability is input or set by a rule: no change from Unreg Flow 2 to Unregulated Spill over timestep, calculate spill based on availability.

2. Availability is 0: no change from Unreg Flow 2 to Unregulated Spill, calculate spill based on Unregulated Spill Table.

3. Availability is greater than 0, check if failure from Unreg Flow 2 to Unregulated Spill occurs during the timestep.

During the third case, the spill over the timestep is calculated using the respective table and multiplied by the availability to find the total spill over the timestep. The total spill is used to predict the pool elevation at the end of the timestep. The calculated pool elevation is compared to the Unreg Flow 2 Failure Elevation. If a failure from Unreg Flow 2 to Unregulated Spill is found to occur any time during the timestep, the failure time is recorded and the ending pool elevation is re-calculated to account for the change in spill due to the change from Unreg Flow 2 to Unregulated Spill during the timestep.

Note: If a failure from Unreg Flow 2 to Unregulated Spill occurs during a dispatch, the first time of this failure is used for the remainder of the dispatch.

The time of failure during the timestep is used to determine what portion of the timestep needs interpolation from each of the two spill tables.

23.1.9.3 Three Unregulated Flows

When the **Three Unregulated Flows** is selected, flow over the spillway is a function of the **Unreg Flow 3 Spill Table.** If the pool elevation meets or exceeds the **Unreg Flow 3 Failure Elevation**, flow becomes a function of the **Unreg Flow 2 Spill Table.** If the pool elevation meets or exceeds the **Unreg Flow 2 Failure Elevation**, flow becomes a function of the **Unreg Flow Spill Table.** To summarize, Unreg Flow 3 fails first, then Unreg Flow 2 fails next as the pool rises. Therefore, the Unreg Flow 2 Failure Elevation should be higher than the Unreg Flow 3 Failure Elevation.

Note: This method was originally called the **Flashboards and Superboards** method but was renamed to be more general. Flashboards and superboards are wooden boards installed in the unregulated spillway so that the reservoir may store more water than what the spillways themselves would allow. The superboards can only be installed if the flashboards are in place.

☛ Unreg Flow Avail and Failure Time

Type: Agg Series Slot
Units: FRACTION
Description: Availability and failure time of Unreg flow 2 and 3 Spill tables.
Information:
I/O: Optional: Availability may be input by user or set by a rule. Failure time is output only
Links: Not linkable

☛ Unreg Flow 3 Failure Elevation

Type: Table Slot
Units: LENGTH
Description: Pool Elevation at which Unreg Flow 3 Spill Table is no longer used.
Information: The value in this slot should be lower than the value in the Unreg Flow 2 Failure Elevation.
I/O: Required Input
Links: Not linkable

☛ Unreg Flow 3 Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding unregulated flow 3 spill values.
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required Input
Links: Not linkable

☛ Unreg Flow 2 Failure Elevation

Type: Table Slot
Units: LENGTH
Description: Pool Elevation at which Unreg Flow 2 Spill Table is no longer used.
Information: The value in this slot should be higher than the value in the Unreg Flow 2 Failure Elevation.
I/O: Required Input
Links: Not linkable

☛ Unreg Flow 2 Spill Table

Type: Table Slot
Units: Length vs. Flow
Description: Pool Elevation vs. corresponding unregulated flow 2 spill values.
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required Input
Links: Not linkable

Method Details

To determine the spill tables to use during a timestep based on availability of Unreg Spill 2 and Unreg Spill 3, the following four possibilities are checked:

1. Both availabilities are input or set by a rule: no change in unregulated spill type over timestep, calculate spill based on availability.
2. One availability is input or set by a rule: Error, both or none must be input.
3. Availabilities are 0. No change in unregulated spill type over timestep. Calculate spill based on Unregulated Spill Table.
4. At least one availability is greater than 0, check for failure in unregulated spill type.

During the fourth case the spill over the timestep is calculated using the respective table and multiplied by the availability to find the total spill over the timestep. The total spill is used to project the pool elevation at the end of the timestep. The calculated pool elevation is compared to the failure elevations of Unreg Flow 2 or Unreg Flow 3. If failure in unregulated spill type is found to occur any time during the timestep, this failure time is recorded and the ending pool elevation is recalculated to account for the change in spill due to the change in unregulated spill type during the timestep.

Note: If a change in unregulated spill type occurs during a dispatch, the first time of this change is used for the remainder of the dispatch.

The time at which change in unregulated spill type occurred during the timestep is used to determine what portion of the timestep needs interpolation from each of the three spill tables.

23.1.10 Regulated Spill Overflow

The category, **Regulated Spill Overflow,** is added if one of the following “regulated” spill methods is selected:

- **Regulated Spill**
- **Regulated and Unregulated**
- **Regulated and Bypass**
- **Regulated, Bypass and Unregulated**
- **Bypass, Regulated and Unregulated**

23.1.10.1 None

This is the default, no-action method.

23.1.10.2 Closed Gate Overflow

This method models the uncontrolled flow over a closed regulated spill gate. This functionality uses the **Regulated Spill Capacity Fraction** to compute the default amount of spillway that is overtopped.

This functionality only applies to **Regulated Spill**, not **Bypass**.

THE FOLLOWING SLOTS WILL BE ADDED:

☛ **CLOSED GATE OVERFLOW**

Type: Series Slot
Units: FLOW
Description: Uncontrolled portion of the Regulated spill that overtops the gates.
Information: This value is computed by the regulated spill method as the value found on the **Closed Gate Overflow Table** multiplied by the **Closed Gate Overflow Capacity Fraction**
I/O: Output only
Links: Not linkable

☛ **CLOSED GATE OVERFLOW TABLE**

Type: Table Slot
Units: LENGTH VS FLOW
Description: Pool Elevation vs unregulated flow.
Information: This table is used to specify the rating curve for uncontrolled flow over the closed Regulated Spill gates. The values should be input to the table as though every Regulated Spill gate is closed. The tables would start with zero flow at or just below the top of the closed gates.
I/O: Input Only

Links: Not available

☛ CLOSED GATE OVERFLOW CAPACITY FRACTION

Type: Series Slot
Units: FRACTION
Description: The fraction of the closed gate overflow that is available.
Information: The value must be between 0.0 and 1.0, inclusive. If not input or set by a rule, it defaults to (1 - **Regulated Spill Capacity Fraction**). Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be set to 0.875 and the Closed Gate Overflow Capacity Fraction would default to 0.125.
I/O: Input, set by a rule, or output
Links: Not linkable

METHOD DETAILS

If not input or set by a rule, **Closed Gate Overflow Capacity Fraction** defaults to (1 - **Regulated Spill Capacity Fraction**). This default indicates that the overflow only happens over gates that are closed. Otherwise, the user can specify the **Closed Gate Overflow Capacity Fraction** slot to say how much of the overflow structure is available.

When the reservoir is below the top of the gates, there is no **Closed Gate Overflow**. But once the reservoir is above the top of one or more closed gates, there is **Closed Gate Overflow**. The computation of this overflow is similar to the unregulated spill computation:

1. A temporary variable called "initHW" is created to represent the Pool Elevation at the beginning of the timestep. Likewise, "endHW" is created to represent the Pool Elevation at the end of the timestep (If the Pool Elevation at the end of the timestep is not known, endHW is set equal to the Pool Elevation at the beginning of the timestep.)

2. The "Closed Gate Overflow Crest" is found from the Closed Gate Overflow Table. It is the Pool Elevation that corresponds to an overflow of zero.

3. If both initHW and endHW are less than or equal to the Closed Gate Overflow Crest, Closed Gate Overflow is set equal to zero.

4. If both initHW and endHW are greater than the Closed Gate Overflow Crest, the average Pool Elevation is used to determine the Closed Gate Overflow from the Closed Gate Overflow Table.

$$\text{Closed Gate Overflow} = \text{Value from table} \times \text{Closed Gate Overflow Capacity Fraction}$$

5. If either initHW or endHW is greater than the Closed Gate Overflow Crest and the other is lower than the crest, the following evaluations and computations are performed:

$$\text{maxHW} = \text{the greater of initHW and endHW}$$

$$\text{minHW} = \text{the lesser of initHW and endHW}$$

$$\text{avgHW} = \frac{\text{maxHW} + \text{Closed Gate Overflow Crest}}{2}$$

$$\text{overflow fraction} = \frac{\text{maxHW} - \text{Closed Gate Overflow Crest}}{\text{maxHW} - \text{minHW}}$$

where:

maxHW = the maximum value of Pool Elevation over the timestep.

minHW = the minimum value of Pool Elevation over the timestep.

avgHW = the average Pool Elevation causing overflow over the timestep.

overflow fraction = corresponds to the fraction of the timestep during which overflow occurs.

A temporary variable called “temp overflow” is obtained from the linear interpolation of the Closed Gate Overflow Table using avgHW. Closed Gate Overflow is then calculated as:

$$\text{Closed Gate Overflow} = \text{overflow fraction} \times \text{temp overflow} \times \text{Closed Gate Overflow Capacity Fraction}$$

When allocating spills to various structures, the **Closed Gate Overflow** must occur at the same time as unregulated spills (i.e. before regulated or bypass). Then any remaining outflow can go through the regulated and or bypass spill structures. Therefore, the minimum regulated spill is computed as follows:

$$\text{Min Regulated Spill} = \text{value from Closed Gate Overflow Table} \times \text{Closed Gate Overflow Capacity Fraction}$$

Also, the **Closed Gate Overflow** is set equal to the Min Regulated Spill.

The functionality assumes that water is either flowing through the gate or over topping it, but not both. The method assumes that the Gate Overflow Table is fixed, that is the elevations in the table do not change. Thus if you had a gate stuck with 1/2 ft open at the bottom and there was still water going over the top, the table (which assumes the gate is closed) would be an incorrect rating.

23.1.11 Input Outflow Adjustment

This method category is only available if a method is selected in the Spill Calculation category. Its purpose is to adjust input Outflow values if they violate a physical constraint.

23.1.11.1 None

This is the default method. It performs no calculations and there are no slots associated with it. The Outflow values will not be adjusted if this method is selected.

23.1.11.2 Reduce Input Outflow

This method is used to reduce the input Outflow value whenever it exceeds the maximum reservoir outflow (due to outlet works capacity).

Slots Specific to this Method

☛ **Requested Outflow**

Type: Series Slot
Units: FLOW
Description: The Outflow value before being adjusted
Information: This slot is available so that the user can see when an Outflow value is adjusted. The value in this slot is the outflow value before being adjusted. A value exists in this slot only if the Outflow value is adjusted.
I/O: Output only
Links: Not linkable

If the Outflow slot value is greater than the maximum reservoir outflow, this method saves the Outflow value in the Requested Outflow slot. Then, the Maximum Capacity flag is set on the Outflow slot. The reservoir is then forced to re-dispatch with the Outflow set to Max Capacity (instead of the original, input value). When the reservoir solves the second time, it computes the maximum reservoir outflow and sets this value on the Outflow slot. The Maximum Capacity flag remains on the Outflow slot for the timestep in question (and will be saved with the model file).

23.1.11.3 Allow Excess Specified Outflows

This method allows input Outflows that exceed the maximum reservoir outflow (due to outlet capacity). Because the excess is above the maximum possible for the (Turbine) Release and Spill slots, it will not be classified as either. Instead, the excess is stored on a separate series slot for reporting or tracking.

Slots Associated With this Method:

☛ OUTFLOW EXCEEDING MAX

Type: Series Slot
Units: FLOW
Description: The portion of the input Outflow that exceeds the sum of the Spill and (Turbine) Release.
Information: This slot tracks the amount that does not fit through the Release and Spill structures.
I/O: Output Only
Links: Not Linkable

METHOD DETAILS:

Toward the end of each dispatch method, if Outflow is greater than the sum of Spill and (Turbine) Release, the Spill method is executed again to redistribute the Outflows to the appropriate spill structures. Within the Spill method, if there is still **no** room for the specified Outflow, the selected method in the **Input Outflow Adjustment** category is executed. When the **Allow Excess Specified Outflows** method is selected, it does the following:

If the Outflow does not have an input flag (I or Z), then the method exits and issues an error that there are excess outflows.

If the Outflow is input (I or Z flag), the method computes the difference between the specified Outflow and the maximum Outflow (i.e. Turbine Release + Spill). This excess outflow is then set on the **Outflow Exceeding Max** slot.

$$\text{Outflow Exceeding Max}[\ ] = \text{Outflow}[\ ] - ((\text{Turbine})\ \text{Release}[\ ] + \text{Spill}[\ ])$$

The method then exits successfully and returns to the Spill method and then the dispatch method. The dispatch method sets the spill and mass balance slots.

23.1.12 Future Value

The methods in this category are used to determine the future value of the energy that would have been generated by the water that was lost through the spillway.

23.1.12.1 None

None is the default method for the Future Value category. No calculations are performed by this method. There are no slots specifically associated with this method.

23.1.12.2 Cumulative Storage Value Table

SLOTS SPECIFIC TO THIS METHOD

☛ **MARGINAL STORAGE VALUE TABLE**

Type: Table
Units: VOLUME VS. $PER ENERGY
Description: Storage versus marginal value per unit energy
Information: This table should be increasing in storage, and usually decreasing in marginal value.
I/O: Required input
Links: Not linkable

☛ **SPILL COST**

Type: Series Slot
Units: $
Description: Future cost of energy lost due to spilled water
Information:
I/O: Output only
Links: May be linked to the Spill Cost slot on the Thermal Object.

☛ **FUTURE VALUE OF USED ENERGY**

Type: Series Slot
Units: $
Description: Future value of energy used in the current timestep
Information:
I/O: Output only
Links: May be linked to Future Value of Used Energy Slot on the Thermal Object.

☛ **ANTICIPATED STORAGE**

Type: SeriesSlot

Units: VOLUME
Description: The combination of Storage in the reservoir at the given timestep plus any flow (converted volume) that is in transit to the reservoir
Information: This slot represents the storage including any lagged flows that are already in a linked upstream reach, which will reach the reservoir at a later timestep. If there are no lagged reaches between this reservoir and the next upstream reservoir, Anticipated Storage will equal Storage. It is this storage value that will be used to calculate Cumulative Storage Value.
I/O: Output only
Links: May be linked

CUMULATIVE STORAGE VALUE

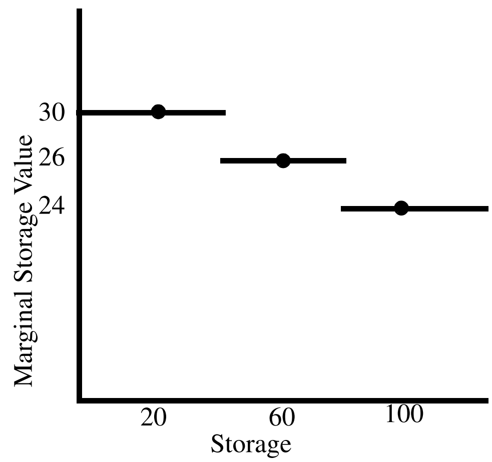
Type: Series Slot
Units: $
Description: Represents the future energy value of the current Anticipated Storage
Information:
I/O: Output only
Links: May be linked to the Total Cumulative Storage Value Slot on the Thermal Object

CUMULATIVE STORAGE VALUE TABLE

Type: Table
Units: VOLUME VS. $
Description: Anticipated Storage and cumulative value used to calculate the Cumulative Storage Value as a function of Anticipated Storage
Information: This table should be increasing in storage and usually increasing in cumulative storage value.
I/O: Required Input either by the user or automatically generated by **RiverWare™** if the Cumulative Storage Value Table Automation method is selected.
Links: Not linkable

This method uses the Marginal Storage Value Table and the calculated Spill and Turbine Release to compute the Spill Cost and Future Value of Used Energy. It then uses the Cumulative Storage Value Table and the calculated Anticipated Storage to compute the Cumulative Storage Value.

The correct marginal value is found from the current storage in the reservoir. If the current storage is than the midpoint between the first and second storage table values, the first marginal value is used. The second marginal value is used for a current storage above that midpoint to the midpoint between the second and third storage table values. The last marginal value is used for any current storage above the midpoint between the second-to-last and the last storage table value. An example is shown in Table 21 on page 1189



Storage	Marginal Value
20	30
60	26
100	24

Table: 21 Marginal Value Table

Assume that the current storage is 39. Therefore, this method would use 30 as the marginal value for use in the next computation. Assume that the current storage is 41. Therefore, this method would use 26 as the marginal value for use in the next computation.

Use of a table in this fashion is unique to this method.

Spill Cost is computed by the following equation:

$$\text{Spill Cost} = \text{Spill} \times \text{Marginal Storage Value} \times \text{Timestep Length}$$

Future Value of Used Energy is computed by the following equation:

$$\text{Future Value of Used Energy} = \text{Turbine Release} \times \text{Marginal Storage Value} \times \text{Timestep Length}$$

The Cumulative Storage Value computation begins by first calculating Anticipated Storage. This is the sum of the reservoir Storage plus any flow already in transit to the reservoir in an upstream lagged reach. For example, assume a reservoir's Inflow slot is linked to a reach with a 3-hour lag time. In an hourly run, the reservoir's Anticipated Storage would be calculated as:

$$\text{Reservoir.Anticipated Storage} = \text{Reservoir.Storage} + (\text{Reach.Inflow(-2)} + \text{Reach.Inflow(-1)} + \text{Reach.Inflow}) \times TimestepLength$$

If there are no lagged reaches between the reservoir and the next upstream reservoir, then Anticipated Storage will simply equal Storage.

The Cumulative Storage Value is then computed by interpolating from the Cumulative Storage Value Table using the calculated Anticipated Storage value.

23.1.13 Cumulative Storage Value Table Automation

This category allows the **RiverWare**™ simulation to automate the creation of the Cumulative Storage Value Table. This category is only visible if Cumulative Storage Value Table is selected in the Future Value category.

23.1.13.1 None

If this method is selected, no automation will be performed and the user must enter the data into the Cumulative Storage Value Table.

23.1.13.2 Marginal Value to Table

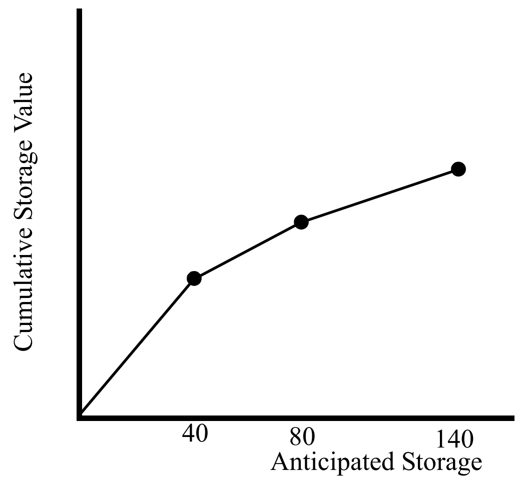
If this method is selected, the Marginal Storage Value table will be used as the source for the generation of the Cumulative Storage Value Table. This is the only calculation associated with this method. There are no slots associated specifically with this method.

This method uses information from the simulation slot Marginal Storage Value Table to generate a cumulative storage value table. The cumulative storage value can be thought of as the summation of the marginal storage values from a storage of 0 to the current storage. Therefore, the automation method finds the same midpoint values used by the simulation Future Value Calc method, and uses those points in the table.

If only one value exists in the Marginal Storage Value Table, then only two entries will exist in the Cumulative Storage Value Table. The two entries will be 0, and midway between the value in the Marginal Storage Value Table, and the maximum value set on the Storage slot. If more than one value exists, three or more points will result. An example is shown below of a Marginal Value Table and the resulting Cumulative Storage Value Table with a graph of the Cumulative Storage Value data.

Storage	Marginal Value
20	30
60	26
100	24

Table: 22 Marginal Value Table



Anticipated Storage	Cumulative Value
40	1200
80	2240
140	3680

Table: 23 Cumulative Storage Value Table

23.1.14 Ramping

This category allows you to model the cost of turbine ramping, the cost of changing the turbine release from one timestep to the next.

23.1.14.1 None

This is the default method for this category. No new slots are instantiated, and no calculations are performed.

23.1.14.2 Track Ramping

In this method the user assigns a unit cost to turbine ramping, a cost per unit change in flow from one timestep to the next. The method then calculates the change in turbine release and the associated ramping cost. The unit cost is the same for ramping up and ramping down.

Slots Specific to this Method

☛ **Ramping Cost**

Type: Series
Units: VALUE
Description: The total cost of turbine ramping for the timestep.
Information: This is the Unit Ramping Cost multiplied by the change in turbine release from one timestep to the net.
I/O: Output only
Links: May be linked

☛ **Turbine Decrease**

Type: Series
Units: FLOW
Description: The difference between Turbine Release at the previous timestep and the current timestep when Turbine Release decreases from the previous timestep
Information: If Turbine Release at the current timestep is greater than at the previous timestep, the value is zero.
I/O: Output only
Links: May be linked

☛ **Turbine Increase**

Type: Series
Units: FLOW
Description: The difference between Turbine Release at the current timestep and the previous timestep when Turbine Release increases from the previous timestep

Information: If Turbine Release at the current timestep is less than at the previous timestep, the value is zero.
I/O: Output only
Links: May be linked

☛ Unit Ramping Cost

Type: Table 1x1
Units: Value per Flow
Description: The cost per unit change in Turbine Release from one timestep to the next
Information: There is a single value for Unit Ramping Cost (i.e. the same unit cost for ramping up and ramping down)
I/O: Optional input; if not input or negative, defaults to zero
Links: Not linkable

The method first checks for a value in the Unit Ramping Cost table slot. If there is no input value, or if the value is negative, it sets the value to zero. Then it checks if Turbine Release is valid for the current timestep and the previous timestep. If Turbine Release at the current timestep is greater than or equal to the previous timestep then the following values are set:

$$\text{Turbine Increase} = \text{Turbine Release} - \text{Turbine Release}(-1)$$

$$\text{Turbine Decrease} = 0$$

$$\text{Ramping Cost} = \text{Unit Ramping Cost} \times \text{Turbine Increase}$$

Otherwise:

$$\text{Turbine Decrease} = \text{Turbine Release}(-1) - \text{Turbine Release}$$

$$\text{Turbine Increase} = 0$$

$$\text{Ramping Cost} = \text{Unit Ramping Cost} \times \text{Turbine Decrease}$$

If Turbine Release is not valid at both the current and previous timesteps, then Turbine Increase, Turbine Decrease and Ramping Cost will all display NaN.

23.1.15 Hydrologic Inflow

The Hydrologic Inflow category allows **RiverWare™** to accommodate inflows to a Reservoir that are not part of the main channel and/or are not gauged. The user methods in this category may be used to initialize the Hydrologic Inflow slot if it is required by the user. If the Hydrologic Inflow slot has been initialized, it is figured into the mass balance when the object dispatches.

23.1.15.1 None

None is the default method for the Hydrologic Inflow category. No calculations are performed by this method. There are no slots specifically associated with this method. If this method is selected, the Hydrologic Inflow slot is not initialized so it is no included in the mass balance.

23.1.15.2 Input Hydrologic Inflow

The Input Hydrologic Inflow method should be used when the user wishes either to input the values of Hydrologic Inflow or have the values default to zero. **RiverWare™** will not overwrite any user input values.

Slots Specific to this Method

☛ **HYDROLOGIC INFLOW**

Type: SeriesSlot
Units: FLOW
Description: flow into the reservoir that is not gauged and/or does not enter through the main channel.
Information:
I/O: Optional; defaults to zero if not input.
Links: Usually input or calculated but could be linked to the Outflow of any object or any other series slot.

☛ **HYDROLOGIC INFLOW ADJUST**

Type: SeriesSlot
Units: FLOW
Description: optional adjustment that can be made to the calculated Hydrologic Inflow
Information:
I/O: Optional; set to zero if not input by the user.
Links: Not linkable

☛ **Hydrologic Inflow Net**

Type: SeriesSlot
Units: FLOW
Description: sum of hydrologic Inflow and Hydrologic Inflow Adjust
Information:
I/O: Output only
Links: Not linkable

The algorithm used for this method is very simple. If Hydrologic Inflow is not input by the user, it is set equal to zero. Hydrologic Inflow Net is calculated as the sum of Hydrologic Inflow and Hydrologic Inflow Adjust.

23.1.15.3 Forecast Hydrologic Inflow

This method is used to forecast the hydrologic inflow based on known inflow values. When this method is selected, the Generate Forecast Hydrology category becomes visible. Within this category, methods are available to generate the hydrologic inflow forecast.

23.1.16 Generate Forecast Hydrology

This category contains methods that forecast the hydrologic inflow based on known inflow values. The user inputs the historical inflows to the reservoir and the methods adjust those values to represent a forecast. The methods in this category execute at the beginning of each timestep.

23.1.16.1 None

This method is the default for this category. It will result in an error if it is selected and a run is started.

23.1.16.2 Geometric Recession

On each timestep in the forecast period, this method will adjust the inflow hydrographs. If the Forecast Period method is selected for the Incremental Hydrologic Inflows on Subbasin category, the Cumulative Hydrologic Inflow is used to forecast and set the Temp Forecasted Cumulative Hydrologic Inflow. If the Forecast Period method is not selected, the input Deterministic Incremental Hydrologic Inflow slot is used to forecast and set the Hydrologic Inflow Forecast series slot.

Slots Specific to this Method

☛ Hydrologic Inflow Forecast

Type: SeriesSlot
Units: FLOW
Description: The forecasted hydrologic inflow values
Information:
I/O: Output only
Links: Not linkable

☛ Deterministic Incremental Hydrologic Inflow

Type: SeriesSlot
Units: FLOW
Description: This slots holds a timeseries of the actual hydrologic inflows to the reach. These values are then adjusted by the forecast method and set on the Hydrologic Inflow Forecast slot.
Information: At the end of the run, the Hydrologic Inflow Forecast slot will be identical to this slot. If the Full Run method is selected, values from the Incremental Hydrologic Inflow slot will be copied into this slot prior to the forecast. This slot is not used (i.e. inputs are ignored) if the Forecast Period method is selected .
I/O: Input or set to the values in the Incremental Hydrologic Inflow slot

Links: Not linkable

☛ Period of Perfect Knowledge

Type: Scalar
Units: FLOW
Description: Number of timesteps for which the forecast will equal the Deterministic Incremental Hydrologic Inflow, i.e., the forecast is known with complete certainty.
Information: Minimum value of 1; maximum value equal to the number of timesteps in the forecast period.
I/O: Input only
Type: Not linkable

☛ Recession Factor

Type: Scalar
Units: NONE
Description: A decimal value that is multiplied by the previous Hydrologic Inflow Forecast value to determine the current value after the Period of Perfect Knowledge.
Information:
I/O: Input only
Links: Not linkable

This method uses different slots and sets different slots if the Forecast Period is selected for the Incremental Hydrologic Inflows on Subbasin category. But, there is always a Source slot and a Target slot where the Source slot is input and used to forecasted and set the Target slot. If the Forecast Period is selected, the computational subbasin calls this method and uses the Cumulative Hydrologic Inflow (Source) to forecast and set the Temp Forecasted Cumulative Hydrologic Inflow (Target). If it is not selected, the reservoir calls this method at the beginning of the timestep and uses the Deterministic Incremental Hydrologic Inflow (Source) slot to forecast and set the Hydrologic Inflow Forecast slot (Target). The following uses the Source/Target terminology to describe the methods.

The Source slot values are required inputs for each timestep. At the beginning of each controller timestep, the Geometric Recession method is executed. For each forecast timestep within the period of perfect knowledge, the Target is set to the Source value. For each forecast timestep after the period of perfect knowledge, the Target is set by multiplying the value of the Target from the previous timestep by the constant recession factor.

A value for the Source slot must be known at every timestep during the run. If Target values are desired past the end of the run, there must also be values in the Source slot at timesteps past the end of the run. If values for Source slot are not entered past the end of the run, the Target values for these timesteps are assumed to be zero.

23.1.16.3 Exponential Recession

On each timestep in the forecast period, this method will adjust the inflow hydrographs. If the Forecast Period method is selected for the Incremental Hydrologic Inflows on Subbasin category, the Cumulative Hydrologic Infow is used to forecast and set the Temp Forecasted Cumulative Hydrologic Inflow. If the Forecast Period methods is not selected, the input Deterministic Incremental Hydrologic Inflow slot is used to forecast and set the Hydrologic Inflow Forecast series slot.

SLOTS SPECIFIC TO THIS METHOD

☛ HYDROLOGIC INFLOW FORECAST

Type: SeriesSlot
Units: FLOW
Description: The forecasted inflow values computed from the Deterministic Incremental Hydrologic Inflow
Information:
I/O: Output only
Links: Not linkable

☛ DETERMINISTIC INCREMENTAL HYDROLOGIC INFLOW

Type: SeriesSlot
Units: FLOW
Description: This slots holds a timeseries of the actual hydrologic inflows to the reach. These values are then adjusted by the forecast method and set on the Hydrologic Inflow Forecast slot.
Information: At the end of the run, the Hydrologic Inflow Forecast slot will be identical to this slot. If the Full Run method is selected, values from the Incremental Hydrologic Inflow slot will be copied into this slot prior to the forecast. This slot is not used (i.e. inputs are ignored) if the Forecast Period method is selected.
I/O: Input or set to the values in the Incremental Hydrologic Inflow slot
Links: Not linkable

☛ FORECAST PERIOD

Type: Table
Units: NONE
Description: Number of timesteps, not including the current timestep, that the inflow hydrograph will be adjusted.
Information:
I/O: Input only
Links: Not linkable

☛ Period of Perfect Knowledge

Type: ScalarSlot
Units: FLOW
Description: Number of timesteps for which the forecast will equal the Deterministic Incremental Hydrologic Inflow, i.e., the forecast is known with complete certainty.
Information: Minimum value of 1; maximum value equal to the number of timesteps in the forecast period.
I/O: Input only
Links: Not linkable

☛ Minimum Forecasted Flow

Type: SeriesSlot
Units: FLOW
Description: The minimum forecasted flow.
Information: If the computed value for Hydrologic Inflow Forecast is less than the Minimum Forecasted Flow, it is set to the Minimum Forecasted Flow.
I/O: Input only
Links: Not linkable

☛ Low Flow Threshold

Type: ScalarSlot
Units: FLOW
Description: The flow rate that dictates whether to use the Low Flow Recession Coefficient or the High Flow Recession Coefficient.
Information:
I/O: Input only
Links: Not linkable

☛ Low Flow Recession Coefficient

Type: ScalarSlot
Units: NONE
Description: The recession coefficient used when the Deterministic Incremental Hydrologic Inflow (at the end of the Period of Perfect Knowledge) is below or equal to the Low Flow Threshold.
Information:
I/O: Input only
Links: Not linkable

☛ High Flow Recession Coefficient

Type: ScalarSlot

Units: NONE
Description: The recession coefficient used when the Deterministic Incremental Hydrologic Inflow (at the end of the Period of Perfect Knowledge) is above the Low Flow Threshold.
Information:
I/O: Input only
Links: Not linkable

This method uses different slots and sets different slots if the Forecast Period is selected for the Incremental Hydrologic Inflows on Subbasin category. But, there is always a Source slot and a Target slot where the Source slot is input and used to forecasted and set the Target slot. If the Forecast Period is selected, the computational subbasin calls this method and uses the Cumulative Hydrologic Infow (Source) to forecast and set the Temp Forecasted Cumulative Hydrologic Inflow (Target). If it is not selected, the reservoir calls this method at the beginning of the timestep and uses the Deterministic Incremental Hydrologic Inflow (Source) slot to forecast and set the Hydrologic Inflow Forecast slot (Target). The following uses the Source/Target terminology to describe the methods.

The Source slot values are input for each timestep. At the beginning of each controller timestep, the Exponential Recession method is executed. For each forecast timestep within the period of perfect knowledge, the Target is set to the Source value. For each forecast timestep after the period of perfect knowledge, the Target slot is set as described below:

$$ForecastedFlow = MAX\left[MinimumForecastedFlow, \left(Source \bullet e^{\frac{(-C)t}{T}}\right)\right]$$

where Source is the value in the Source slot at the end of the period of perfect knowledge, C is the recession coefficient, t is the elapsed time of the forecast period, and T is the total time from the end of the period of perfect knowledge to the end of the forecast period.

If the Source at the end of the period of perfect knowledge is negative, the Target at that timestep is exactly equal to the Source. However, the Source used in the recession equation, is the last positive value for Source. In the event that there is not a positive value for the Source, RiverWare issues a warning, and all values for Hydrologic Inflow Forecast within the forecast period will be set to the Minimum Forecasted Flow.

A value for the Source slot must be known at every timestep during the run. If the Target values are desired past the end of the run, there must also be values in the Source slot at timesteps past the end of the run. If values for Source are not entered past the end of the run, the Target values for these timesteps are assumed to be zero.

23.1.16.4 Coefficient and Exponent

On each timestep in the forecast period, this method will adjust the inflow hydrographs. If the Forecast Period method is selected for the Incremental Hydrologic Inflows on Subbasin category, the Cumulative Hydrologic Infow is used to forecast and set the Temp Forecasted Cumulative Hydrologic Inflow. If the Forecast Period methods is not selected, the input Deterministic Incremental Hydrologic Inflow slot is used to forecast and set the Hydrologic Inflow series slot.

Slots Specific to this Method

☛ Hydrologic Inflow

Type: SeriesSlot
Units: FLOW
Description: flow into the reservoir that is not gaged and/or does not enter through the main channel.
Information:
I/O: Output only
Links: Not linkable

☛ Hydrologic Inflow Adjust

Type: SeriesSlot
Units: FLOW
Description: optional adjustment that can be made to the calculated Hydrologic Inflow
Information:
I/O: Optional; the Hydrologic Inflow Adjust may either be input by the user or it is set to zero if it is not input.
Links: Not linkable

☛ Hydrologic Inflow Net

Type: SeriesSlot
Units: FLOW
Description: Sum of Hydrologic Inflow and Hydrologic Inflow Adjust
Information:
I/O: Output only
Links: Not linkable

☛ Forecast Inflow Parameters

Type: Table
Units: NONE
Description: Table slot that contains four parameters used in the forecast inflow method. The first row contains the values for the increasing hydrograph, the second row contains values for the decreasing hydrograph. The first column contains coefficients, the second column contains exponents.
Information: 2X2 table

I/O: Input only

Links: not linkable

☛ Deterministic Incremental Hydrologic Inflow

Type: SeriesSlot

Units: FLOW

Description: This slots holds a timeseries of the actual Hydrologic Inflows to the reach. These values are then adjusted by the forecast method and set on the Hydrologic Inflow slot.

Information: At the end of the run, the Hydrologic Inflow Forecast slot will be identical to this slot. If the Full Run method is selected, values from the Incremental Hydrologic Inflow slot will be copied into this slot prior to the forecast. This slot is not used (i.e. inputs are ignored) if the Forecast Period method is selected.

I/O: Input or set to the values in the Incremental Hydrologic Inflow slot

Links: not linkable

☛ Forecast Period

Type: Table

Units: NONE

Description: Number of timesteps, not including the current timestep, that the inflow hydrograph will be adjusted.

Information:

I/O: Input only

Links: not linkable

This method uses different slots and sets different slots if the Forecast Period is selected for the Incremental Hydrologic Inflows on Subbasin category. But, there is always a Source slot and a Target slot where the Source slot is input and used to forecasted and set the Target slot. If Forecast Period is NOT selected, the reservoir calls this method at the beginning of the timestep and uses the Deterministic Incremental Hydrologic Inflow (Source) slot to forecast and set the Hydrologic Inflow slot (Target). If it is selected, the computational subbasin calls this method and uses the Cumulative Hydrologic Infow (Source) to forecast and set the Temp Forecasted Cumulative Hydrologic Inflow (Target). The subbasin then computes the incremental flow and sets the value on the Hydrologic Inflow slot. The following description uses the Source/Target terminology to describe the method.

The method works as follows: on the current timestep, the Target is set equal to the Source. The method then loops through the remaining timesteps in the forecast period and sets the Target using the following formula starting at i = 1:

$$HI_i = HI_{i-1} + HI_{i-1}\frac{(KI_i - KI_{i-1})}{KI_{i-1}}((C^i)^E)$$

where KI_i is the Source at timestep i, HI_i is the Target at timestep i. The counter i represents the timestep beyond the current timestep. For example, i = 1 is the next timestep, i = 2 is the current timestep + 2 timesteps, etc. The coefficient, C, and exponent, E, are the values in the Forecast Inflow Parameters slot. If(Source(i-1) <= Source(i)), E and C are the increasing (rising) values. Otherwise, E and C are the decreasing (falling) values.

In the above formula, there is a mathematical problem if KI_{i-1} is zero. In this situation, the Target at that index is set to the known inflow at that index. This allows the simulation to continue with reasonable values for the Target.

At the end of the method, the Hydrologic Inflow Net is set equal to the Hydrologic Inflow plus the Hydrologic Inflow Adjust for each timestep in the loop. This allows the rules to be able to use the Hydrologic Inflow Net before the object dispatches (such as in the GetMaxOutflowGivenInflow() function).

A value for the Source slot must be known at every timestep during the run. If forecasted Hydrologic Inflow values are desired past the end of the run, there must also be values in the Source slot at timesteps past the end of the run. If values for Source are not entered past the end of the run, the Hydrologic Inflows for these timesteps are assumed to be zero.

23.1.17 Incremental Hydrologic Inflows on Subbasin

The Incremental Hydrologic Inflows on Subbasin category contains methods used to specify that the reservoir has cumulative inflows that must be disaggregated into incremental inflows. There are two methods: Full Run and Forecast Period. The disaggregation is actually executed from the computation subbasin containing the reservoir. For more information, click HERE (Section 7.1.22).

When the Input Hydrologic Inflow method is selected in the Hydrologic Inflow category, the Full Run becomes available. The method contains two slots: Cumulative Hydrologic Inflow and Incremental Hydrologic Inflow for the reservoir. Data must be input into the Cumulative Hydrologic Inflow slots. The computational subbasin will execute the Compute Full Run Incremental Hydrologic Inflows method and set the Incremental Hydrologic Inflow slots as input. Setting the Incremental Hydrologic Inflow slots as input prevents the slot values from being cleared in future model runs when the subbasin is disabled. The computational subbasin will first check that Full Run method has been selected on all reservoirs and reservoirs in the basin. During the model run, when the reservoir dispatches, the Full Run method will copy the value in the Incremental Hydrologic Inflow slot and set the Hydrologic Inflow slot.

When a forecast method is selected in the Generate Forecast Hydrology methods on the reservoir, a new method will be available within the Incremental Hydrologic Inflows on Subbasin category: Forecast Period on the reservoir. This method will contain the Cumulative Hydrologic Inflow slot. When this method is selected the computational subbasin will execute the Compute Forecast Period Incremental Hydrologic Inflows method and set the Hydrologic Inflow slots.

23.1.17.1 None

This method is the default for the Incremental Hydrologic Inflows on Subbasin category and should be selected when hydrologic inflow data is not cumulative or the computation of incremental hydrologic inflows is not desired. There are no slots specifically associated with this method.

23.1.17.2 Full Run

The Full Run method is available from the Incremental Hydrologic Inflows on Subbasin category on the reservoirs (storage reservoir, level power reservoir, and sloped power reservoir,). This method is only available if the Input Hydrologic Inflow method is selected in the Hydrologic Inflow category. This method holds the slots necessary for the computation of incremental hydrologic inflows that is performed by the computational subbasin. The method contains two slots: Cumulative Hydrologic Inflow and Incremental Hydrologic Inflow. These slots will be accessed by the computational subbasin when executing the Compute Full Run Incremental Local Inflows method. The computational subbasin uses input Cumulative Hydrologic Inflow values to calculate and set the

Incremental Hydrologic Inflow slot. Refer to the computational subbasin's Incremental Local Inflows documentation for details, click HERE (Section 7.1.22.2). The Full Run method on the reservoir copies the results of the calculation in the Incremental Hydrologic Inflow slot values over to the Hydrologic Inflow series slot.

Slots Specific to this Method

☛ **Cumulative Hydrologic Inflow**

Type: SeriesSlot
Units: Flow
Description: The cumulative hydrologic inflow to the reservoir
Information: Hydrologic inflow is cumulative either (1) between headwater control points and the first reservoir in the river system and also throughout the system between two reservoirs, or (2) throughout the entire river system.
I/O: Required Input
Links: Not linkable

☛ **Incremental Hydrologic Inflow**

Type: SeriesSlot
Units: Flow
Description: The incremental hydrologic inflow to the reservoir
Information: This slot is set by the Calculate Incremental Flows method on the computational subbasin and represents the actual hydrologic inflow to the reservoir. If a Generate Forecast Hydrology method is selected, the Deterministic Incremental Hydrologic Inflow slot will be set to the values in this slot.
I/O: Computed and set with the Input flag
Links: Usually not linked

23.1.17.3 Forecast Period

The Forecast Period method is available from the Incremental Hydrologic Inflows on Subbasin category on the reservoir. This method is only available if the Forecast Hydrologic Inflows method is selected in the Hydrologic Inflow category and one of the forecasting methods (i.e. Geometric Recession, Exponential Recession, or Coefficient and Exponent) is selected in the Generate Forecast Hydrology category. If this method is selected, but the reservoir is not part of a subbasin with the appropriate methods selected, an error will be issued. The method contains two slots: Cumulative Hydrologic Inflow and Temp Forecasted Cumulative Hydrologic Inflow. These slots will be accessed by the computational subbasin when executing the Compute Forecast Period Incremental Hydrologic Inflows method. The computational subbasin will use the user input Cumulative Hydrologic Inflow value to forecast and set the Temp Forecasted Cumulative Hydrologic Inflow slot. It then uses this temporary value in its calculation of the incremental flows. The final result of this method (forecasted incremental hydrologic inflows) is set on the slot Hydrologic Inflow Forecast for each timestep in the forecast period. (Note: Hydrologic Inflow is set instead if the Coefficient and Exponent method is selected). Refer to the computational subbasin's Incremental Local Inflows documentation for details, click HERE (Section 7.1.22.3).

SLOTS SPECIFIC TO THIS METHOD

☛ CUMULATIVE HYDROLOGIC INFLOW

Type: SeriesSlot
Units: FLOW
Description: The cumulative hydrologic inflow to the reservoir
Information: Hydrologic inflow is cumulative either (1) between headwater control points and the first reservoir in the river system and also throughout the system between two reservoirs, or (2) throughout the entire river system.
I/O: Required Input
Links: Not linkable

☛ TEMP FORECASTED CUMULATIVE HYDROLOGIC INFLOW

Type: SeriesSlot
Units: FLOW
Description: the forecasted cumulative Hydrologic inflow to the control point
Information: This slot is set by the selected Generate Forecast Hydrologic Inflows method on the control point as called from the computational subbasin. It represents the cumulative Hydrologic inflow to the control point forecasted throughout the forecast period. This slot is a temporary slot and is not saved with the model file.
I/O: Output only
Links: Usually not linked

23.1.18 Slope Storage

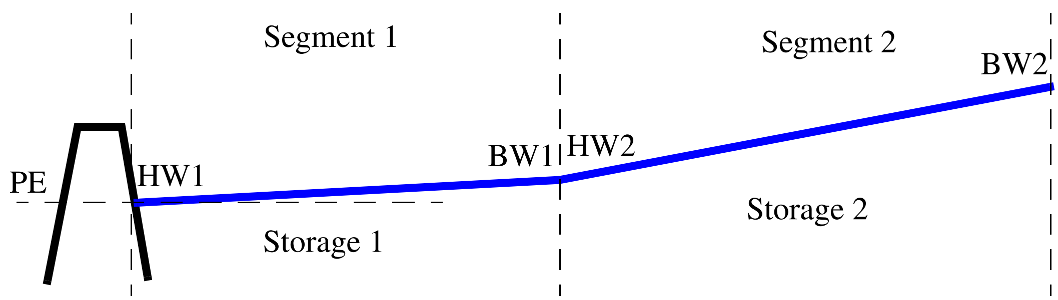
Slope Power Reservoirs contain a portion of their Storage in the wedge created by the sloping water surface. The following user method calls to additional methods that perform the mass balance calculations in Slope Power Reservoirs for all dispatch methods.

23.1.18.1 None

This is the default method in the Slope Storage category. It performs no calculations and has no slots associated with it. An error is posted if this method is selected. The user is required to choose Storage Segments.

23.1.18.2 Storage Segments

The user has the option of dividing a reservoir into longitudinal segments. This is most useful for large reservoirs and relatively small timesteps, where the propagation of a change in flow takes more than one timestep to travel the length of the reservoir. Reservoir segments are numbered in the upstream direction, with the first segment being the farthest downstream.



The **Storage Segments** method does not perform any calculations. Instead, this method makes visible the new method category, **Slope Storage Coefficients**. You then have the option of selecting one of two methods, **Impulse Response** method or **Weighting Coefficients** method, to calculate the segment storage. But the **Storage Segments** method requires you to specify the number of segments in the reservoir and provide the headwater, backwater, and storage relationships for the **Segments in Reservoir** that are used by the Slope Storage Coefficient methods to calculate segment storage. Also, you can specify the **Segment Storage Adjustment** based on the backwater elevation and change in backwater elevation.

Slots Specific to this Method

☛ Segments in Reservoir

Type: Table
Units: NONE
Description: number of longitudinal segments in the reservoir

Information: Multiple segments are useful for large reservoirs and relatively small timesteps, where the propagation of a change in flow takes more than one timestep to travel the length of the reservoir. Reservoir segments are numbered in the upstream direction, with the first segment being the farthest downstream.

I/O: Required input

Links: Not linkable

☛ PROFILE STORAGE TABLE

Type: Table

Units: LENGTH VS. LENGTH VS. VOLUME

Description: total storage of each segment corresponding to a headwater and a backwater elevation

Information: 3-D table containing data relating headwater and backwater elevations to segment storage. Each reservoir segment has its own three-column block of data. Data must be entered into these tables in a structured way in order for the 3-dimensional table interpolator to work correctly. Headwater must be input in blocks of the same values, increasing down the table. For every headwater elevation block in column 1, backwater values should be listed in increasing order in column 2, with the corresponding segment storage in column 3. A sample table for a one-segment reservoir is shown below:

Headwater	Backwater	Storage
500	500	800
500	510	805
500	520	810
550	550	900
550	560	905
550	570	910
600	600	1000
600	610	1010
600	620	1020

I/O: Required input

Links: Not linkable

☛ SEGMENT STORAGE

Type: Aggregate Series

Units: VOLUME

Description: This slot holds the storage in each segment of the reservoir. The total storage in the reservoir is equal to the sum of the segment storage (within convergence).

Information: There will be one column for each segment. The number of columns is automatically adjusted at run start.

I/O: Output only

Links: Not Linkable

☛ SEGMENT STORAGE ADJUSTMENT

Type: Table Slot

Units: LENGTH, LENGTH VOLUME

Description: This 3D table will have one 3 column block for each segment in the reservoir. The first column of the block is the upstream elevation of the segment, the second column is the change in elevation from the previous timestep, and the third column is the storage adjustment factor.

Information: The table must have 1 block for each segment. This will be adjusted at run start. It can have NaNs at the bottom of the table. If there is no data in first row for a segment, then no adjustment will be applied to that segment.The correction can be either positive or negative. The table below shows an example correction table. The correction for Segment 1 is not applied (i.e. always zero), but Segment 2 does have a correction factor.

Segment 1			Segment 2		
Upstream Elevation 1 ft	Upstream Elevation Change 1 ft	Storage Adjustment 1 cfs-day	Upstream Elevation 2 ft	Upstream Elevation Change 2 ft	Storage Adjustment 2 cfs-day
NaN	NaN	NaN	550	-4	10
NaN	NaN	NaN	550	-2	2
NaN	NaN	NaN	550	0	0
NaN	NaN	NaN	550	2	-2
NaN	NaN	NaN	550	4	-11
NaN	NaN	NaN	575	-4	9
NaN	NaN	NaN	575	-2	2
NaN	NaN	NaN	575	0	0
NaN	NaN	NaN	575	2	-3
NaN	NaN	NaN	575	4	-12

I/O: Input only.

Links: Not linkable.

23.1.19 Slope Storage Coefficients

Slope Power Reservoirs contain a portion of their Storage in the wedge(s) created by the sloping water surface. The visibility of the **Slope Storage Coefficients** is dependent upon selecting the **Storage Segments** from the **Slope Storage**. The **Slope Storage Coefficients** contains two methods: **Impulse Response** and **Weighting Coefficients**. Both methods are iterative and based on mass balance calculations and data tables describing relationships between steady flow, headwater elevation, backwater elevation, and storage on the reservoirs.

23.1.19.1 Impulse Response

This is the default method.

An iterative method based on mass balance calculations and data tables describing relationships between steady flow, headwater elevation, backwater elevation, and storage on reservoirs. The data tables are based on a steady state backwater profile. The four mass balance variables are Inflow, Outflow, Storage and Pool Elevation (Pool Elevation refers to the elevation at the dam, also called the headwater elevation). Two of these must be known to calculate the other two. The method solves for Storage and either Inflow, Outflow, or Pool Elevation by iterating until a calculated mass balance storage agrees with a Storage value looked up from tables created from the steady flow, headwater elevation, backwater elevation, and storage relationships. If both Inflow and Outflow are known, a convergence algorithm that is detailed HERE (Appendix A: Reservoir Convergence) is used to calculate storage.

First, a mass balance storage is calculated with known and estimated parameters (Inflow, Outflow, Previous Storage). Then, a second Storage value is found with a table look up.

Flow Param is the steady flow used to determine headwater, backwater, and storage from the relationships given by the data tables. It is calculated with the following equation:

$$FlowParam(i) = \frac{K(i) \times [a(i) \times segInflow(i) + b(i) \times segOutflow(i)]}{2.0}$$

where i is the reservoir segment. K(i) is a calibration parameter dependent on previous Storage. segInflow(i) and segOuflow(i) are the calculated inflow and outflow of each segment and are local variables. Weighting factors, a(i) and b(i), determine to what extent Flow Param depends on the inflow and outflow of the segment.

SegInflow and segOutflow are calculated using reservoir inflows and outflows at current and previous timesteps weighted with impulse response coefficients (IRC). IRCs represent the effect of current and previous flow values on the current segment flow. IRCs are specified by the user in the Profile Coeff Table. For each IRC, a flow fraction is calculated equal to a flow value multiplied by the IRC. The first IRC is multiplied by the current Inflow (or Outflow) value, and each additional IRC is multiplied by a previous flow value. For cases where current or previous flow values are not known, the most recent flow estimate is used on the first iteration. SegInflow (or segOutflow) is equal to the sum of these flow fractions.

Flow Param and either a given or estimated Pool Elevation (headwater) are then used to look up a backwater elevation on the Profile Backwater Table. Then a storage value is looked up from the Profile Storage Table using the headwater and backwater elevations to get the segment storage. Then a 3 dimensional lookup is performed on the **Segment Storage Adjustment** table slot. It uses the computed backwater elevation and the change in backwater elevation (current computed value minus previous value) to get the storage adjustment. The segment storage is adjusted by this value and stored in an array. After the total storage is found, the **Slope Storage** method sets the **Segment Storage** from the array values.

This total storage value is compared to the calculated mass balance storage. The method iterates, by adjusting the unknown parameter (Inflow, Outflow, or Pool Elevation), until the two storage values are within convergence.

Local inflows contributing to wedge storage are represented in the slot Prof Hydro Inflow. If not specified by user, Prof Hydro Inflow is set to the value of Hydrologic Inflow Net. Setting Prof Hydro Inflow gives the user an opportunity to specify part of the local inflows contributing to the wedge and part to the level storage. The amount of Prof Hydro Inflow contributing to each reservoir segment is controlled by the wc parameter, which is specified in the Profile Coeff Table. The Prof Hydro Inflow is multiplied by wc before it is added to segInflow.

Slots Specific To This Method

☛ Flow Param

Type: Agg Series Slot
Units: FLOW
Description: flow which produces a steady state profile to give correct headwater, backwater, and storage relationships for each segment
Information: This slot contains a column for each reservoir segment. A user-specified value overrides the formula calculation.
I/O: Optional; if not set by user, it is calculated with segment inflow, segment outflow, and weighting coefficients (see above description).
Links: Not linkable

☛ Prof Hydro Inflow

Type: Agg Series Slot
Units: FLOW
Description: local inflow to a slope power reservoir that contributes to the wedge storage
Information: Prof Hydro Inflow is multiplied by the wc parameter before being added to segInflow. It contains a column for each reservoir segment.
I/O: Optional; if not set by the user, it will be set to the Hydrologic Inflow at that timestep. If it is not set by the user and None is the selected method for Hydrologic Inflow, it defaults to zero.

Links: Not linkable

☛ INTERMED BACKWATER ELEV

Type: Table Series
Units: LENGTH
Description: the water surface elevation at the upstream end of each reservoir segment
Information: Contains a column for each reservoir segment.
I/O: Output only
Links: Not linkable

☛ PROFILE COEFF TABLE

Type: Table
Units: NONE
Description: coefficients used in determining a flow parameter for each segment
Information: Contains a, b, wc, and impulse response coefficients for each reservoir segment. a and b are coefficients weighting the contribution of segment inflow and segment outflow to the calculation of Flow Param. The parameter, wc, is a coefficient adjusting the amount of Prof Hydro Inflow contributing to the segment inflow. Impulse response coefficients are used to calculate flow fractions that are summed to determine the segment inflow or segment outflow. The first coefficient is multiplied by a current flow value and the subsequent coefficients are multiplied by previous flow values to calculate flow fractions. The sum of the impulse response coefficients for each segment usually equal 1.0. A sample Profile Coeff Table is shown below:
I/O: Required input
Links: Not linkable

	Segment 1	Segment 2
a	0.7	2.0
b	1.3	0.0
wc	0.5	0.0
Impulse Response Coefficients	0.5	1.0
	0.5	0.0

☛ PROFILE BACKWATER TABLE

Type: Table
Units: FLOW VS. LENGTH VS. LENGTH

Description: backwater elevation of each segment corresponding to a flow parameter and a headwater elevation

Information: 3-D table containing data relating a steady flow (Flow Param) to a headwater elevation and a backwater elevation. Each reservoir segment has its own three-column block of data. Data must be entered into these tables in a structured way in order for the 3-dimensional table interpolator to work correctly. Flow Param must be input in blocks of the same values, increasing down the table. For every Flow Param block in column 1, headwater values should be listed in increasing order in column 2, with the corresponding backwater in column 3. A sample table for a one-segment reservoir is shown below:

Flow Param	Headwater	Backwater
100	500	505
100	550	555
100	600	605
200	500	510
200	550	560
200	600	610
300	500	515
300	550	565
300	600	615

I/O: Required input

Links: Not linkable

☛ Profile K Coeff Table

Type: Table

Units: VOLUME vs. NONE

Description: K parameter associated with a change in storage from one and two previous timesteps

Information: Allows the user to adjust the K parameter, a calibration factor used in calculating Flow Param. Its value is set depending on whether Storage is suddenly increasing or decreasing. This table contains two columns for each reservoir segment. The first column is a delta Storage value representing the change in storage between one and two previous timesteps. Delta storage can be either positive or negative. The second column is the K value corresponding to a given delta storage.

I/O: Required input

Links: Not linkable

23.1.19.2 Weighting Coefficients

The user has the option of dividing a reservoir into longitudinal segments and further dividing those segments into longitudinal partitions. This is most useful for large reservoirs and relatively small timesteps, where the propagation of a change in flow takes more than one timestep to travel the length of the reservoir. Reservoir partitions are numbered in the upstream direction, with the first partition being the farthest downstream. Several slots require or provide information for each reservoir partition.

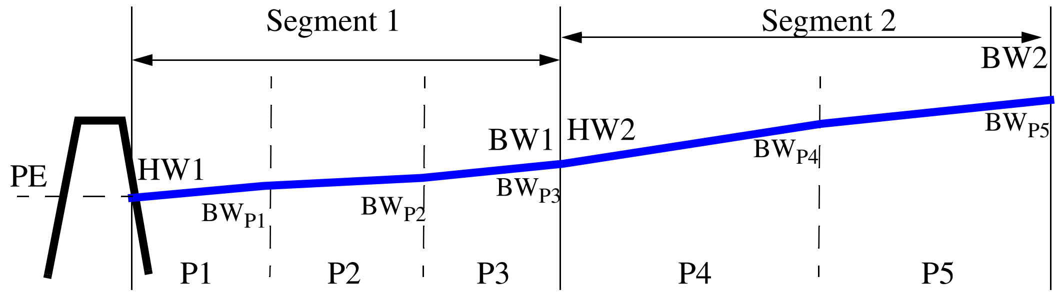
This is an iterative method based on mass balance calculations and data tables describing relationships between steady flow, headwater elevation, backwater elevation, and storage of the partition. The four mass balance dispatch variables are Inflow, Outflow, Storage and Pool Elevation (Pool Elevation refers to the elevation at the dam, also called the headwater elevation). Two of these must be known to calculate the other two. The method solves for Storage and either Inflow, Outflow, or Pool Elevation by iterating until a calculated mass balance storage agrees with a Storage value looked up from tables created from the steady flow, headwater elevation, backwater elevation, and storage relationships for the segments. Partitions are not included in the storage iterations. If both Inflow and Outflow are known, a convergence algorithm that is detailed **HERE (Appendix A: Reservoir Convergence)** is used to calculate storage.

The **Weighting Coefficients** method performs a check at the beginning of the run to make sure the user has populated the **Partition Profile Coef Table** with input. The user has the option of specifying the **Partition Flow Parameters** or calculating it with the weighting coefficients. If calculated, the **Partition Flow Parameters** at each partition is a function of the weighting coefficients particular to that partition as input by the user in the **Partition Profile Coef Table**. The **Partition Flow Parameters** slot is set to the result of the below equation, which describes how the flow parameter at each partition is calculated. The partition flow parameter “P_j(t)” is calculated at each partition “j”, where the time “t” ranges from the current timestep to some previous timestep “n” as:

$$\begin{aligned} P_j(t) \;=\; K \times [\, & a_{j1}I(t) + a_{j2}I(t-1) + \ldots + a_{jn}I(t-n) \\ & + b_{j1}H(t) + b_{j2}H(t-1) + \ldots + b_{jn}H(t-n) \\ & + c_{j1}I2(t) + c_{j2}I2(t-1) + \ldots + c_nI2(t-n) \\ & + d_{j2}P_j(t-1) + d_{j3}P_j(t-2) + \ldots + d_{jn}P_j(t-n) \\ & + e_{j1}O(t) + e_{j2}O(t-1) + \ldots + e_{jn}O(t-n)] \end{aligned}$$

Where, K is the value obtained by looking up the previous change in storage (i.e., Storage[t-1] minus Storage[t-2]) on the appropriate segment’s columns of the **Profile K Coeff Table**. I(t) is the inflow to the reservoir; a_{jt} is the coefficient weighting the contribution of inflow at the partition “j” at timestep “t”. H(t) is the hydrological inflow; b_{jt} is the coefficient weighting contribution of hydrological inflow at partition “j” at timestep “t”. The calculation for the Partition Flow Parameter uses the Hydrologic Inflow Net if a hydrologic inflow exists. If there is no hydrologic inflow, it is not included in the calculations of the Partition Flow Parameter. I2(t) corresponds to the Inflow 2; c_{jt} is its weighting coefficient similar to a and b. O(t) is the outflow from the reservoir; e_{jt} is its weighting coefficient. Finally d_{jt} is the weighting coefficient corresponding to previous timestep's Partition Flow Parameter, P_{jt}. The number of previous timesteps included in the calculation for the Partition Flow Parameter is determined by the user as the number of rows appended in the **Partition Profile Coef Table** (that contains the weighting coefficients) by the user.

The following diagram can be used to visualize the order in which the partitions will solve for their BW elevations.



The partitions are the areas between the breaks as denoted above. For the first partition, P1, the headwater at the dam will be utilized in conjunction with the flow parameter calculated for P1 in a 3D table interpolation from the user input **Partition BW Table** to find the BW elevation (at the upstream end) of P1. The headwater elevation at the dam PE/HW1 will also be used to calculate the BW_{P1}, BW_{P2} and BW_{P3}. P3 also constitutes the end of segment 1. Thus, HW2 is set to BW_{P3}. Then, HW2 is used to calculate BW_{P4}. HW2 will be used to find the BW elevations for each partition in segment 2. The BW elevation of the last partition in a segment is the HW in the next segment and will be used to calculate all the BW elevations of all the partitions in the segment.

The storage calculations of the segments follow: 1) The backwater elevation for segment 1, BW1 is set to the BW_{P3} 2) Likewise, the backwater elevation for segment 2, BW2 is set to the BW_{P5}. The storage for segment 1 is calculated using the headwater at the dam HW1 and BW1 via a 3D table interpolation from the **Profile Storage Table**. Then a 3 dimensional lookup is performed on the **Segment Storage Adjustment** table slot. It uses the computed backwater elevation BW1, and the change in backwater elevation (current computed value, BW1, minus previous value, BW1(t-1)) to get the storage adjustment. The segment storage is adjusted by this value and is stored in an array.

The storage for segment 2 is calculated using HW2 and BW2 via a 3D table interpolation from the **Profile Storage Table**. The **Segment Storage Adjustment** is applied for this segment and the Segment Storage is stashed in the array. The method then sums the storages of each segment. After the total storage is found, the function returns to the Slope Storage method and sets the Segment Storage from the array values and the iteration continues as described above.

SLOTS SPECIFIC TO THIS METHOD

Partitions per Segment

Type: Table
Units: None
Description: Table Slot that contains the number of partitions in each longitudinal segment of Slope Power Reservoir
Information: The number of each partitions in each segment can be input individually for that particular segment. If the user does not input the number of partitions in each segment or an individual segment, the number of partitions defaults to 0. The number of partitions specified must be an integer.
I/O: Input Optional
Links: Not linkable

Partition Profile Coef Table

Type: Table
Units: None
Description: Table Slot that contains the user specified weighting coefficients for each partition to be used in calculating the **Partition Flow Parameters**.
Information: The number of columns in this slot is determined by the total number of partitions as specified in the **Partitions per Segment** slot. Each partition will generate five columns for the weighting coefficients for that partition, which correspond to: Inflow, Hydrologic Inflow, Inflow2, Partition Flow Parameters at earlier time steps, and Outflow from the reservoir. The number of rows corresponds to the number of previous time steps the user wishes to use in calculating the Partition Flow Parameter. A sample **Partition Profile Coef Table** for one partition is shown below. The coefficient D should be zero for t. Otherwise, it will use previously computed Partition Flow Parameters for the current timestep which are from a previous iteration and would not be meaningful.
I/O: Required Input
Links: Not linkable

	Parameter:	A1	B1	C1	D1	E1
	Multiplier:	Inflow	Hydrologic Inflow	Inflow 2	Partition Flow Parameter	Outflow
Time	t	0.11	0.07	-0.061	0	0.061
	t-1	0.07	0.02	0	0.66	0
	t-2	0	0.045	0	0	0

Partition BW Table

Type: Table

Units: FLOW VS. LENGTH VS. LENGTH
Description: Backwater elevation of each segment corresponding to a flow parameter and a headwater elevation.
Information: 3-D table containing data relating a steady flow to a headwater elevation and a backwater elevation. Each partition has its own three column block of data. Data must be entered into the table in a structured manner so that the 3-D table interpolation will work correctly. Flow parameters must be input in blocks of the same values, increasing down the table. For every flow parameter block in column 1 of the partition, headwater values should be listed in increasing order in column 2, with the corresponding backwater elevation in column 3. A sample table for one partition is shown below.
I/O: Required Input
Links: Not linkable

Flow Param 1	Headwater 1	Backwater 1
100	500	505
100	550	555
100	600	605
200	500	510
200	550	560
200	600	610
300	500	515
300	550	565
300	600	615

☛ PARTITION BW TABLE AUTO MAX

Type: Table
Units: FLOW VS LENGTH FOR EACH PARTITION
Description: Automatically generated table relating Flow Parameter to the maximum possible Headwater.
Information: This table is created at the beginning of the run by taking each Flow Param from the Partition BW Table and finding the largest Headwater. This is done for each partition. This table is then used in the iterative slope storage computations (like RPL functions or GetMaxOutGivenInflow) as a bound. Within these computations, if the Flow Parameter or Headwater exceeds the table values, then the values are set to the maximum and the iteration continues. This allows intermediate computations to succeed when they would have otherwise stopped the run. An error is issued if the final result is outside the table limits.
I/O: Output - Read only

Links: Not Linkable

☛ PARTITION BW TABLE AUTO MIN

Type: Table
Units: FLOW VS LENGTH FOR EACH PARTITION
Description: Automatically generated table relating Flow Parameter to the minimum possible Headwater.
Information: This table is created at the beginning of the run by taking each Flow Param from the Partition BW Table and finding the smallest Headwater. This is done for each partition. This table is then used in the iterative slope storage computations (like RPL functions or GetMaxOutGivenInflow) as a bound. Within these computations, if the Flow Parameter or Headwater is below the table values, then the values are set to the minimum and the iteration continues. This allows intermediate computations to succeed when they would have otherwise stopped the run. An error is issued if the final result is outside the table limits.
I/O: Output - Read only
Links: Not Linkable

☛ PARTITION FLOW PARAMETERS

Type: Agg Series Slot
Units: FLOW
Description: Flow that produces a steady state profile to give correct headwater and backwater relationships for each partition.
Information: This slot contains a column for each partition as set in the **Partitions per Segment** Slot.
I/O: Optional, if not input by user, will be calculated using Inflow from and upstream dam, Hydrologic Inflow, Inflow2, previous Partition Flow Parameters, Outflow, and the user specified weighting coefficients from the **Partition Profile Coef Table**. Note, the column labels of this slot can be modified to be more intuitive.
Links: Not Linkable

☛ PARTITION BW ELEVATION

Type: Table Series
Units: LENGTH
Description: Backwater elevation at each partition
Information: This slot contains a column for each partition. It is solved through a 3-D table interpolation between the **Partition Flow Parameters** slot and the **Partition BW Table**. Note, the column labels of this slot can be modified to be more intuitive.
I/O: Output only

Links: Not linkable

☛ Profile K Coeff Table

Type: Table
Units: VOLUME vs. NONE
Description: K parameter associated with a change in storage from one (t-1) and two (t-2) previous timesteps.
Information: Allows the user to adjust the K parameter, a calibration factor used in calculating Flow Param. Its value is set depending on whether Storage is suddenly increasing or decreasing. This table contains two columns for each reservoir segment. The first column is a delta Storage value representing the change in storage between one and two previous timesteps. Delta storage can be either positive or negative. The second column is the K value corresponding to a given delta storage.
I/O: Required input
Links: Not linkable

23.1.20 Target Slope Storage

Used to solve target operations. The methods solve a target Storage used for computing a lumped mass balance over several time periods.

23.1.20.1 None

This is the default method which performs no calculations. It is only used if no target operations are being executed. There are no slots associated with this method.

23.1.20.2 Lumped Mass Balance

This method must be selected whenever a target operation is performed on a reservoir. It is only available when the Storage Segments method is selected. The method finds a Storage value at the target time based on Outflow and Pool Elevation using an iterative routine similar to one of the Slope Storage Coefficients methods. This storage value is used to compute a lumped mass balance across the target operation time range. There are no slots specifically associated with this method.

23.1.21 Evaporation and Precipitation

The Evaporation and Precipitation category methods are used to calculate the volume of Evaporation from and Precipitation to the surface of a reservoir over the timestep. Precipitation and Evaporation are used in the mass balance equations which are solved in the dispatch methods.

Some of the methods in this category only calculate evaporation.

23.1.21.1 None

The None method is the default method for the Evaporation and Precipitation category. It should be chosen if the user does not want to include Evaporation in the mass balance equation of the Reservoir. There are no slots specifically associated with this method. No calculations are performed by this method.

23.1.21.2 Daily Evaporation

The Daily Evaporation method is used to calculate the daily evaporation volume and the flow rate of the precipitation. The daily evaporation volume is a function of the Evaporation Rate, average Surface Area, and Pan Coefficient.

Slots Specific to this Method

☛ Elevation Area Table

Type: Table Slot
Units: LENGTH vs. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☛ Evaporation

Type: SeriesSlot
Units: VOLUME
Description: volume of water lost to evaporation during one timestep
Information:
I/O: Output only
Links: Not linkable

☛ Evaporation Table

Type: Table Slot
Units: NOUNITS vs. LENGTH (per day)

Description: day of the year vs. Evaporation Rate
Information: The first of January is 0.
I/O: Required Input
Links: Not linkable

☛ Pan Evaporation Coefficient

Type: Table Slot
Units: NO UNITS
Description: a fractional value between 0 and 1 that represents the portion of potential evaporation which actually occurs
Information:
I/O: Required Input
Links: Not linkable

☛ Precipitation Rate

Type: SeriesSlot
Units: LENGTH VS. TIME
Description: precipitation intensity for the given timestep
Information:
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

☛ Precipitation Volume

Type: SeriesSlot
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☛ Surface Area

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area calculated from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

The Surface Area is determined using the Pool Elevation and the Elevation Area Table. The Evaporation Rate is looked up in the Evaporation Table according to the current day of the year. Evaporation is calculated using the following equation:

$$\text{Evaporation} = \text{Evaporation Rate} \times \text{Pan Evaporation Coefficient} \times (\text{Surface Area} + \text{Surface Area (-1)})/2$$

The volume of Precipitation that occurred over the timestep is then calculated with the following equation:

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)})/2$$

where in the above equations:

Evaporation Rate = the Evaporation Rate corresponding to the current day of the year

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

23.1.21.3 Input Evaporation

The Input Evaporation method should be used when the user wants to input the Evaporation Rate directly. This Evaporation Rate is used to compute the volume of water that evaporated over the timestep. Also, the user can input the evaporation volume directly on the Evaporation slot. In that case, the Evaporation Rate is not used to calculated Evaporation.

Slots Specific to this Method

☛ Elevation Area Table

Type: Table Slot
Units: LENGTH vs. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☛ Evaporation

Type: SeriesSlot
Units: VOLUME
Description: volume of water lost due to evaporation during the timestep
Information:
I/O: Output; optional input overrides calculation
Links: Not linkable

☛ EVAPORATION RATE

Type: SeriesSlot
Units: LENGTH PER TIME
Description: rate at which water evaporates from the surface
Information:
I/O: Optional input, disaggregated by method as described in the Evap and Precip Rate Specification category, or defaults to 0.0 if not specified by the user.
Links: Not linkable

☛ PRECIPITATION RATE

Type: SeriesSlot
Units: LENGTH PER TIME
Description: precipitation intensity for a given timestep
Information:
I/O: Optional input, disaggregated by method as described in the Evap and Precip Rate Specification category, or defaults to 0.0 if not specified by the user.
Links: Not linkable

☛ PRECIPITATION VOLUME

Type: SeriesSlot
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☛ SURFACE AREA

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

At the beginning of the run, the chosen method in the **Evap and Precip Rate Specification** category is executed. This category allows you to specify the rates as monthly or periodic slots.

If the user specifies Evaporation directly (via input or rules), the value will be used instead of calculating a value below.

If Evaporation is not specified, the following equation is used to compute the volume of water that evaporated from the Reservoir over the timestep:

$$\text{Evaporation} = \text{Evaporation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)}) / 2$$

The precipitation flow rate over the timestep is calculated as shown in the following equation:

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)}) / 2$$

where in the above equations:

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

23.1.21.4 Monthly Evaporation

In the Monthly Evaporation method, evaporation is calculated linearly from the Evaporation Coefficients entered for each month. This method will not work with a timestep longer than monthly. The total evaporated volume is a function of the average Reservoir Surface Area over the timestep, the Evaporation Coefficient, and the length of the timestep. The following slots are specifically associated with this method.

Slots Specific to this Method

Elevation Area Table

Type: TableSlot
Units: LENGTH vs. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

Evaporation

Type: SeriesSlot
Units: VOLUME
Description: volume of water lost to evaporation during one timestep
Information:
I/O: Output only
Links: Not linkable

Evaporation Coefficients

Type: TableSlot
Units: LENGTH PER TIME

Description: rate of evaporation for each month
Information: This slot contains one column of values. The Evaporation Coefficient for each month of the year must be input by the user beginning with the Evaporation Coefficient for January.
I/O: Required input
Links: Not linkable

☛ Precipitation Rate

Type: SeriesSlot
Units: LENGTH PER TIME
Description: precipitation intensity for the given timestep
Information: Value must be input by the user for each timestep.
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

☛ Precipitation Volume

Type: SeriesSlot
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☛ Surface Area

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area calculated from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

The Surface Area of the Reservoir is calculated based on the Elevation Area Table. The Evaporation is then calculated using the following formula:

$$\text{Evaporation} = \text{Evaporation Coefficient} \times (\text{Surface Area} + \text{Surface Area (-1)})/2 \times TimestepLength$$

The volume of Precipitation that occurred over the timestep is then calculated using the following equation:

$$\text{Precipitation} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)})/2 \times TimestepLength$$

where in the above equations:

Evaporation Coefficient = the Evaporation Coefficient for the current month

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

23.1.21.5 Pan and Ice Evaporation

The Pan and Ice Evaporation method is used to calculate the volume of evaporation with one of two methods based on the value of the Pan Ice Switch slot for each timestep. The Pan Ice Switch slot is used as an indicator of whether ice is present on the surface of the reservoir. A value of 1.0 in the Pan Ice Switch slot indicates that there is ice cover on the Reservoir that must be taken into account when Evaporation is calculated. A value of 0.0 or any number other than 1.0 in the Pan Ice Switch slot indicates that there is no ice on the surface of Reservoir. The following slots are those specifically associated with this method.

Slots Specific to this Method

☛ **Evaporation**

Type: Series
Units: VOLUME
Description: volume of water lost to evaporation during the current timestep
Information:
I/O: Output only
Links: Not linkable

☛ **Elevation Area Table**

Type: Table
Units: LENGTH vs. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☛ **K Factor**

Type: Series Slot with Periodic Input
Units: VELOCITY per TEMPERATURE_F
Description: factor relating average temperature, in degrees Fahrenheit, to evaporation rate
Information: This slot is a series slot, but the data can be input as a periodic relationship.
I/O: Optional but is required Input if the Pan Ice Switch slot is 1.0
Links: Not linkable

☛ Max Air Temperature

Type: Series
Units: TEMPERATURE IN FARENHEIT
Description: maximum air temperature during the timestep
Information:
I/O: Optional; required only if the Pan Ice Switch slot is 1.0
Links: Not linkable

☛ Min Air Temperature

Type: Series
Units: TEMPERATURE IN FARENHEIT
Description: minimum air temperature during the timestep
Information:
I/O: Optional; required if the Pan Ice Switch slot is 1.0
Links: Not linkable

☛ Pan Evaporation

Type: Series
Units: LENGTH PER TIME
Description: evaporation rate from the surface
Information:
I/O: Optional; only required if the Pan Ice Switch is 0.0
Links: Not linkable

☛ Pan Evaporation Coefficient

Type: Table
Units: DECIMAL
Description: weighing factor for pan evaporation rate
Information:
I/O: Optional; required if the Pan Ice Switch slot is 0.0
Links: Not linkable

☛ Pan Ice Switch

Type: Series
Units: No Units
Description: indicator of surface ice coverage for each timestep; **1.0** = ice; any other number or **0.0** = no ice.
Information: This slot is a series slot, but the data can be input as a periodic relationship.
I/O: Required input
Links: Not linkable

☛ Precipitation Rate

Type: Series
Units: LENGTH PER TIME
Description: precipitation intensity for a given timestep
Information:
I/O: Optional; defaults to 0.0 if not specified by the user.
Links: Not linkable

☛ Precipitation Volume

Type: Series
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☛ Surface Area

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

☛ Surface Ice Coverage

Type: SeriesSlot
Units: DECIMAL
Description: fraction of the Surface Area which is covered by ice
Information:
I/O: Optional; only used if the Pan Ice Switch slot is 1.0. Defaults to 0.0 for any timestep not specified by the user.
Links: Not linkable

If the Pan Ice Switch slot is equal to 1.0, ice is present and the following calculation is performed to compute evaporation:

$$\text{Evaporation} = \frac{\text{Max Air Temperature} + \text{Min Air Temperature}}{2} \times \text{K Factor} \times (1 - \text{Surface Ice Coverage}) \times \text{average Surface Area} \times \mathit{Timestep}$$

If the calculated Evaporation is less than zero, the Evaporation is set equal to zero.

The Precipitation is calculated with the following equation if the Pan Ice Switch slot is equal to 1.0:

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times (1 - \text{Surface Ice Coverage}) \times \text{average Surface Area}$$

The volume of precipitation that accumulated over the timestep at the Reservoir (Precipitation Volume) is the product of the precipitation flow rate and the timestep.

If the Pan Ice Switch slot is 0.0 or any number other than 1.0, there is no ice and the following calculation is performed to compute Evaporation:

$$\text{Evaporation} = \text{Pan Evaporation} \times \text{Pan Evaporation Coefficient} \times \text{average Surface Area} \times \text{Timestep}$$

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times \text{average Surface Area}$$

The volume of precipitation that accumulated over the timestep at the Reservoir (Precipitation Volume) is the product of the precipitation flow rate and the timestep.

23.1.21.6 Pan and Ice Evaporation, Current Surface Area

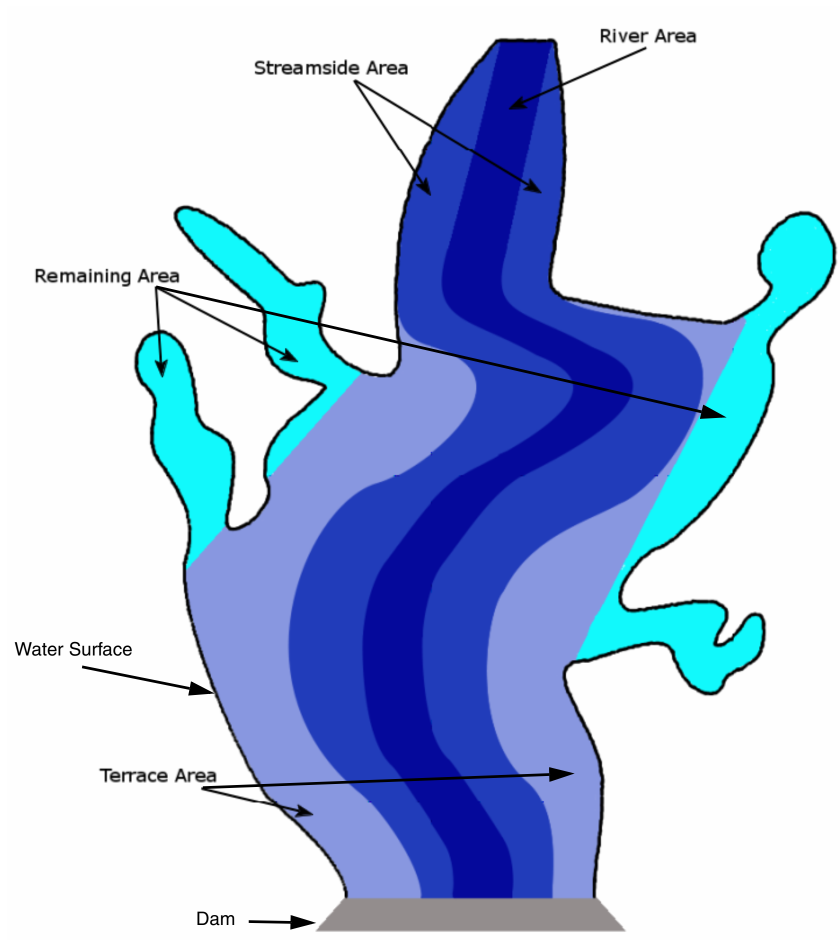
This method is exactly the same as the Pan and Ice Evaporation method. It uses the same slots, has the same required inputs and performs the same calculations. The only difference is that this method uses the instantaneous, end of timestep surface area instead of the average surface area over the timestep.

23.1.21.7 Periodic Net Evaporation

Periodic Net Evaporation computes the gross evaporation from the reservoir and then subtracts out components of evaporation that would have occurred if the reservoir had not been built. This is the net evaporation and is set in the **Evaporation** slot. Each area of the submerged reservoir is separate including:

- River
- Streamside
- Terrace, and any
- Remaining areas

Each can have a separate evaporation coefficient and possibly additional components in its computation like temperature. The area of each region is specified in a separate table relating reservoir pool elevation to each region's area. It is assumed that any precipitation that falls on the Remaining Area would have completely evaporated. The figure shows a sample of the different reservoir areas used in this method.



Slots Specific to this Method

☛ Average Precipitation

Type: Periodic
Units: Velocity (Length per Time)
Description: Slot describing the average precipitation
Information: Typically this would have a yearly period and monthly precipitation values. It is used in the computation of Remaining Evaporation.
I/O: Required Input
Links: Not Linkable

☛ Average Air Temperature

Type: Periodic
Units: TemperatureInFahren
Description: Slot describing the average air temperature

Information: Typically this would have a yearly period and monthly temperature values. It is used in computation of Streamside Evaporation and Terrace Evaporation.
I/O: Required Input
Links: Not Linkable

☛ Elevation Area Table
Type: Table
Units: Length vs Area
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required Input
Links: Not Linkable

☛ Elevation River Area
Type: Table
Units: Length vs Area
Description: Table relating reservoir Pool Elevation to submerged river area.
Information:
I/O: Required Input
Links: Not Linkable

☛ Elevation Streamside Area
Type: Table
Units: Length vs Area
Description: Table relating reservoir Pool Elevation to submerged streamside area.
Information:
I/O: Required Input
Links: Not Linkable

☛ Elevation Terrace Area
Type: Table
Units: Length vs Area
Description: Table relating Pool Elevation to submerged terrace area
Information:
I/O: Required Input
Links: Not Linkable

☛ Evaporation
Type: Series
Units: Volume

Description: Water lost from the reservoir to evaporation. This is the net evaporation and is the value that is included in the reservoir mass balance.
Information: This is the calculated as Gross Evaporation minus Salvage Evaporation
I/O: Output only
Links: Not Linkable

☛ Gross Evaporation
Type: Series
Units: Volume
Description: The total evaporation off the reservoir surface. This is the evaporation that is actually occurring from the reservoir.
Information: This is calculated as GrossEvaporationCoeff times SurfaceAreaAvg converted from a flow to volume.
I/O: Output only
Links: Not Linkable

☛ Gross Evaporation Coefficient
Type: Periodic
Units: Velocity (Length per Time)
Description: A table that describes the gross evaporation coefficient as it varies periodically. This is similar to a pan evaporation coefficient.
Information:
I/O: Required Input
Links: Not Linkable

☛ River Evaporation Coefficient
Type: Periodic
Units: Velocity (Length per Time)
Description: A table that describes the river evaporation coefficient as it varies periodically. This is similar to a pan evaporation coefficient.
Information:
I/O: Required Input
Links: Not Linkable

☛ Salvage Evaporation
Type: Series
Units: Volume
Description: The evaporation that would have occurred if the reservoir were not in place.

$$SalvageEvaporation = RiverEvaporation + StreamsideEvaporation + TerraceEvaporation + RemainingEvaporation$$

Information:
I/O: Output Only
Links: Not Linkable

☛ Streamside Coefficient

Type: Periodic
Units: VelocityPerTemperature_F (i.e. Length per Time per Temperature_F)
Description: Periodic table of coefficients
Information: Typically this represents a unit depth per month per degree Fahrenheit (e.g. inches per month per degree Fahrenheit)
I/O: Required Input
Links: Not Linkable

☛ Surface Area

Type: Series
Units: Area
Description: Reservoir surface area computed from a lookup on the Elevation Area table
Information:
I/O: Output only
Links: Not Linkable

☛ Terrace Coefficient

Type: Periodic
Units: VelocityPerTemperature_F (i.e. Length per Time per Temperature_F)
Description: Periodic table of coefficients
Information: Typically this represents a unit depth per month per degree Fahrenheit (e.g. inches per month per degree Fahrenheit)
I/O: Required Input
Links: Not Linkable

The method will be passed in a current estimate of Surface Area and Average Surface Area. The latter is an average of the current estimate and previous timestep’s value. In this description, it is called SurfaceAreaAvg. Similarly, StreamsideAreaAvg, RiverAreaAvg, TerraceAreaAvg and RemainingAreaAvg are all averages of the current and previous values.

In the following steps, the FlowToVolume and VolumeToFlow notation indicates that the specified expression will be converted from a flow to a volume (or vice versa) using the

timestep length. This is necessary for the units to work correctly as evaporation [Volume units] is computed as a coefficient [Length/Time units] times an area [L^2 units]. Note that in the following steps, the slots are in bold while intermediate values are not.

The method does the following:

1. Get the value from the periodic **Gross Evaporation Coefficient** slot. If not valid, issue an error.

2. Compute **Gross Evaporation**:

$$GrossEvaporation = FlowToVolume(GrossEvaporationCoefficient \times SurfaceAreaAvg)$$

3. Get the value from the periodic **River Evaporation Coefficient** slot. If not valid, issue an error.

4. Look up the Pool Elevation at t and t-1 on the **Elevation River Area** table to get the RiverArea at t and t-1. Then

$$RiverAreaAvg = \frac{RiverArea[t] + RiverArea[t-1]}{2}$$

5. Compute River Evaporation:

$$RiverEvaporation = FlowToVolume(RiverEvaporationCoefficient \times RiverAreaAvg)$$

This simulates that the river evaporation that would have occurred without the reservoir is a function of area and coefficient.

6. Get the value from the periodic **Streamside Coefficient** slot. If not valid, issue an error.

7. Look up the Pool Elevation at t and t-1 on the **Elevation Streamside Area** table to get the StreamsideArea at t and t-1. Then:

$$StreamsideAreaAvg = \frac{StreamsideArea[t] + StreamsideArea[t-1]}{2}$$

8. Get the value from the periodic **AverageTemperature** slot. If not valid, issue an error.

9. Compute Streamside Evaporation:

$$StreamsideEvaporation = FlowToVolume(StreamsideCoefficient \times StreamsideAreaAvg \times AverageAirTemperature)$$

This simulates that the streamside evaporation that would have occurred without the reservoir is a function of area, coefficient, and average air temperature.

10. Get the value from the periodic **Terrace Coefficient** slot. If not valid, issue an error.

11. Look up the Pool Elevation at t and t-1 on the **Elevation Terrace Area** table to get the Terrace Area at t and t-1. Then:

$$TerraceAreaAvg = \frac{TerraceArea[t] + TerraceArea[t-1]}{2}$$

12. Compute Terrace Evaporation:

$$TerraceEvaporation = FlowToVolume(TerraceCoefficient \times TerraceAreaAvg \times AverageAirTemperature)$$

This simulates that the terrace evaporation that would have occurred without the reservoir is a function of area, coefficient, and average air temperature.

13. Compute the average Remaining Area as:

$$RemainingAreaAvg = SurfaceAreaAvg - RiverAreaAvg - StreamsideAreaAvg - TerraceAreaAvg$$

If RemainingAreaAvg is less than zero, an error will be issued as the table data is incorrect.

14. Get the value from the periodic **Average Precipitation**. If not valid, issue an error.

15. Compute RemainingEvaporation:

$$RemainingEvaporation = FlowToVolume(RemainingArea \times AveragePrecipitation)$$

This simulates that all of the precipitation on the Remaining Area would have evaporated.

16. Compute **Salvage Evaporation**:

$$SalvageEvaporation = RiverEvaporation + StreamsideEvaporation + TerraceEvaporation + RemainingEvaporation$$

17. Compute **Evaporation** as follows:

$$Evaporation = GrossEvaporation - SalvageEvaporation$$

The **Evaporation** is then a volume that is removed from the reservoir mass balance in the dispatch method.

23.1.21.8 Single Evaporation

In the Single Evaporation method, evaporation is calculated linearly from the Single Evaporation Coefficient entered by the user. The total evaporated volume is a function of the average Reservoir Surface Area over the timestep, the Single Evaporation Coefficient, and the length of the timestep. The following slots are specifically associated with this method.

Slots Specific to this Method

☛ Elevation Area Table

Type: Table Slot
Units: LENGTH VS. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☛ Evaporation

Type: SeriesSlot
Units: VOLUME
Description: volume of water lost to evaporation during one timestep
Information:
I/O: Output only
Links: Not linkable

☛ Single Evap Coeff

Type: Table Slot
Units: LENGTH PER TIME
Description: rate of evaporation
Information: This slot contains a single value that represents the evaporation rate.
I/O: Required input
Links: Not linkable

☛ Precipitation Rate

Type: SeriesSlot
Units: LENGTH PER TIME
Description: precipitation intensity for the given timestep
Information: Value must be input by the user for each timestep.
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

☛ Precipitation Volume

Type: SeriesSlot
Units: VOLUME

Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

Surface Area

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area calculated from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

The Surface Area of the Reservoir is calculated based on the Elevation Area Table. The Evaporation is then calculated using the following formula:

$$\text{Evaporation} = \text{Evaporation Coefficient} \times (\text{Surface Area} + \text{Surface Area (-1)})/2 \times TimestepLength$$

The volume of Precipitation that occurred over the timestep is then calculated using the following equation:

$$\text{Precipitation} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)})/2 \times TimestepLength$$

where in the above equations:

Evaporation Coefficient = SingleEvapCoeff entered by the user

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

23.1.22 Evap and Precip Rate Specification

This category allows you to choose how the evaporation and precipitation rates will be specified. The category is only available when the **Input Evaporation** method in the **Evaporation and Precipitation** category is specified.

23.1.22.1 None

This is the default method; that is, the rates must be input, set by a rule, or they default to 0.0.

23.1.22.2 Monthly Rates

This method allows you to specify the evaporation and precipitation rates as a series of monthly values for the entire run.

☛ **EVAPORATION RATE MONTHLY**

Type: Series Slot
Units: VELOCITY
Description: The evaporation rate for each month of the run.
Information: You must set the timestep for this series slot to be monthly. Because this slot is monthly, it is most likely different than the run timestep. As a result, if you "synchronize objects", you must select the toggle in the synchronization control to "Exclude Slots with Different Timestep from Run." This will prevent changing the timestep of this slot when other slots are synchronized.
I/O: Optional input
Links: Not linkable

☛ **PRECIPITATION RATE MONTHLY**

Type: Series Slot
Units: VELOCITY
Description: The precipitation rate for each month of the run.
Information: You must set the timestep for this series slot to be monthly. Because this slot is monthly, it is most likely different than the run timestep. As a result, if you "synchronize objects", you must select the toggle in the synchronization control to "Exclude Slots with Different Timestep from Run." This will prevent changing the timestep of this slot when other slots are synchronized.
I/O: Optional input
Links: Not linkable

METHOD DETAILS

At the beginning of run, the method disaggregates the **Evaporation Rate Monthly** and **Precipitation Rate Monthly** to the **Evaporation Rate** and **Precipitation Rate** slots,

respectively. If the timestep of the run is monthly, it uses the values directly. If the timestep of the run is less than a month, it **look ups** the month that contains the given timestep and uses that value. No interpolation is performed.

If the run timestep is annual, an error is issued.

If the two slots are not monthly but have inputs, an error is issued.

If there is no value in the monthly slot for a given month, then the rate is set to 0.0.

23.1.22.3 Periodic Rates

This method allows you to specify the evaporation and precipitation rates as a periodic relationship.

☛ **Evaporation Rate Periodic**

Type: Periodic Slot
Units: Velocity
Description: The evaporation rate as a periodic relationship.
Information: Like other periodic slots, you can choose the period and whether to interpolate or lookup.
I/O: Required Input
Links: Not linkable

☛ **Precipitation Rate Periodic**

Type: Periodic Slot
Units: Velocity
Description: The precipitation rate as a periodic relationship.
Information: Like other periodic slots, you can choose the period and whether to interpolate or lookup. If you do not wish to model precipitation, you still must enter a zero in this periodic slot.
I/O: Required Input
Links: Not linkable

Method Details

At the beginning of run, the method sets the **Evaporation Rate** and **Precipitation Rate** slots by looking up (or interpolating as configured on the periodic slot) the given timestep in the **Evaporation Rate Periodic** and **Precipitation Rate Periodic** slots, respectively. If accessing the periodic slot fails due to missing values, then an error is issued and the run stops.

23.1.23 Low Flow Releases

This category is only used to add the slots necessary for low flow release calculations. These slots are generally used by a RPL function (called MeetLowFlowRequirement) to compute the low flow releases necessary to meet the low flow requirements on control point objects.

23.1.23.1 None

This method performs no calculations and adds no slots.

23.1.23.2 Enable Low Flow Releases

This method performs no calculations. It simply adds the Low Flow Release slot and Maximum Low Flow Delivery Rate slot.

Slots Specific To This Method

☛ Low Flow Release

Type: Series Slot
Units: FLOW
Description: The portion of the Outflow that is intended to meet a low flow requirement
Information: This slot is normally computed by a RPL function (MeetLowFlowRequirement) that computes the low flow releases necessary to meet the low flow requirements on control point objects.
I/O: Usually set by a rule
Links: Not linkable

☛ Maximum Low Flow Delivery Rate

Type: Periodic Slot
Units: FLOW
Description: The maximum low flow delivery rate for the reservoir
Information: This value is used by the RPL function (MeetLowFlowRequirement) that determines the low flow releases from each reservoir. Low flow releases will be limited to this value.
I/O: Required input
Links: Not linkable

23.1.24 Target Operation

The Target Operation category is used to enable algorithms which calculate different lumped mass balance algorithms required by Target Operations (see TARGET flag). None of the methods exist as separate functions, meaning that the target operation algorithms themselves must be used in conjunction with the other mass balance algorithms. It is important to note that either the Simple Target or Lagged Target method must be selected if a target operation is set on the Reservoir.

23.1.24.1 None

This is the default method in the Target Operation category. It performs no calculations. It may only be selected if a Target Operation is not performed on the object. There are no slots specifically associated with this method.

23.1.24.2 Simple Target

The Simple Target method distributes the required Inflow or Outflow evenly among all available timesteps. The total flow is divided among all the non-input timesteps included within the Target Operation to meet the Target. For most cases with a target operation, this is the desired behavior. There are no slots specifically associated with this method.

23.1.24.3 Lagged Target

The Lagged Target method should be used when there are lag times in the upstream reaches. When Inflows are known, this method solves for the same solution as the Simple Target method. When Outflows are known, this method distributes the required Inflow so a steady Outflow occurs from an upstream, non-integer timestep lagged, reservoir. To distribute Inflows, the Total Lag of Upstream Reaches and Outflow from Upstream Reservoir are used to calculate the required Inflows to the current Reservoir. These required Inflows must meet the Target and result in steady Outflow from the upstream Reservoir. The solution yields a perturbed Inflow at the first undetermined timestep of the Target Operation, followed by steady Inflows for the remainder of the undetermined target times. This solution removes the numerical instability which would be produced in the Outflow of the upstream reservoir using the Simple Target method.

Slots Specific to this Method

☛ **Outflow from Upstream Reservoir**

Type: SeriesSlot
Units: FLOW
Description: outflow from the upstream reservoir
Information:
I/O: Required input

Links: Should be linked to the Outflow slot of the upstream reservoir.

☛ Total Lag of Upstream Reaches

Type: Table
Units: TIME
Description: combined lag time of reaches between the current and the upstream reservoir
Information:
I/O: Required input
Links: Not linkable

23.1.25 Sediment

The Sediment category is used to enable algorithms which adjust reservoir Elevation Volume and possibly Elevation Area relationships in response to sediment inflow.

23.1.25.1 None

The None method is the default for the Sediment category. No calculations are performed in this method. There are no slots specifically associated with this method.

23.1.25.2 CRSS Sediment

The **CRSS Sediment** method is designed based on sedimentation calculations performed by the US Bureau of Reclamation's Colorado River Simulation System (CRSS) model. This function distributes reservoir sediment based on the "Empirical Area Reduction Method". Simply put, sediment is distributed through an iterative process in which a total volume loss due to sedimentation is calculated based on an assumed top of sediment elevation.

Slots Specific to this Method

☛ **Elevation Area Table**

Type: Table
Units: LENGTH vs. AREA
Description: generated elevation area table for calculating sediment distribution
Information:
I/O: Output only
Links: Not linkable

☛ **Elevation Vol Area Table Increment**

Type: Table
Units: LENGTH
Description: elevation increments for the generated Elevation Volume and Elevation Area Tables
Information: This table often needs more precise elevation increments than the sediment calculation tables.
I/O: Required input
Links: Not linkable

☛ **Initial Elevation Area Table**

Type: Table
Units: LENGTH vs. AREA
Description: initial elevation area table

Information: Provided for comparison with initial data
I/O: Output only
Links: Not linkable

☛ Initial Elevation Volume Table

Type: Table
Units: LENGTH VS. VOLUME
Description: initial elevation volume table
Information: provided for comparison with initial data
I/O: Output only
Links: Not linkable

☛ Sediment Distribution Coefficients

Type: Table
Units: NOUNITS
Description: parameters for empirical equation governing sediment distribution
Information:
I/O: Required input
Links: Not linkable

☛ Sediment Inflow

Type: SeriesSlot
Units: VOLUME
Description: volume of sediment flowing into the reservoir at each timestep
Information:
I/O: Required input
Links: Not linkable

☛ User Input Elev Area Data

Type: Table
Units: LENGTH VS. AREA
Description: initial Elevation Area relationship
Information: These values are initial conditions for the first timestep of the simulation. The elevation increments will be used for all sedimentation calculations.
I/O: Required input
Links: Not linkable

This volume loss is recalculated (with a new top of sediment elevation) at each iteration, until the calculated volume loss is equal to the actual volume of sediment inflow (within a specified convergence). The total volume loss calculation consists of a somewhat complicated algorithm utilizing elevation/area and elevation/volume data for the reservoir and an empirical equation. The empirical equation uses user specified parameters which relate the portion of total area that is taken up by sediment to the Pool Elevation. The empirical equation basically gives the shape of the accumulated sediment. The empirical equation has a close relationship to the elevation volume and elevation area characteristics of a given reservoir. The elevation/area and elevation/volume data is stored in a polynomial coefficient table, which gets recalculated after each timestep. The actual Elevation Area, Elevation Volume tables used by **RiverWare**™ are adjusted at the end of the sedimentation code (but prior to the hydrologic simulation).

Caution should be exercised in creating input data for this method. The close relationship between the empirical area reduction equation and the shape of the reservoir (reflected in the User Input Elev Area Data) makes the method fairly sensitive to input data. When choosing empirical parameters for this method, physical characteristics of the given reservoir need to be considered. The Bureau of Reclamation currently considers 4 possible types of reservoirs, with each type having a corresponding set of empirical area reduction parameters. The reservoir type classification is based on the shape of the Reservoir, the manner in which the reservoir is to be operated, and the size of the sediment particles to be deposited in the reservoir. The main emphasis is on the shape. Tables are used to classify the reservoirs based on these characteristics. Once the type has been established, the parameter values for that type can also be taken from tables in the literature. An incorrect set of parameters for a given reservoir will lead to an inability to achieve convergence on the sediment distribution within this method.

23.1.26 Bank Storage

The Bank Storage methods are used to calculate the volume of water stored in the Reservoir banks. These methods also calculate the change in the volume of water stored in the Reservoir banks from one timestep to the next.

23.1.26.1 None

None should be chosen if the user does not want to calculate the amount of Bank Storage in the Reservoir. This is the default method for the Bank Storage category. Bank Storage and the Change in Bank Storage are set to zero but are not displayed. There are no slots specifically associated with this method. No calculators are performed in this method.

23.1.26.2 Input Bank Storage

The Input Bank Storage method allows users to directly input values into the Bank Storage slot or to set these values using a rule. Change in Bank Storage is calculated internally in RiverWare for use in the mass balance equations.

SLOTS SPECIFIC TO THIS METHOD

☛ **BANK STORAGE**

Type: Series Slot
Units: VOLUME
Description: volume of water stored in the reservoir banks
Information:
I/O: Input Only
Links: Usually not linked, but could be linked to Data Object.

☛ **CHANGE IN BANK STORAGE**

Type: Series Slot
Units: VOLUME
Description: change in volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

23.1.26.3 CRSS Bank Storage

The CRSS Bank Storage method replicates the U.S. Bureau of Reclamation's CRSS bank storage calculation. The Bank Storage and the Change in Bank Storage are calculated using the Reservoir Storage and the Bank Storage Coefficients.

SLOTS SPECIFIC TO THIS METHOD

☞ BANK STORAGE

Type: Series Slot
Units: VOLUME
Description: volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

☞ BANK STORAGE COEFFICIENT

Type: Table Slot
Units: NO UNITS VS. NO UNITS
Description: gain or loss of storage vs. change in bank storage
Information: The first coefficient (column zero) is for increasing storage and the second coefficient is for decreasing storage.
I/O: Required input
Links: Not linkable

☞ CHANGE IN BANK STORAGE

Type: Series Slot
Units: VOLUME
Description: change in volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

There are two ways Bank Storage can be calculated depending on the current Storage of the Reservoir. If the Reservoir's current Storage is greater than the Reservoir's Storage at the previous timestep, the Storage is increasing. Bank Storage is calculated using the following equation:

$$\text{Bank Storage} = \text{Bank Storage(-1)} + (\text{first Bank Storage Coefficient} \times (\text{Storage} - \text{Storage(-1)}))$$

If the Reservoir's current Storage is less than the Reservoir's Storage at the previous timestep, the Storage is decreasing. Bank Storage is calculated using the following equation:

$$\text{Bank Storage} = \text{Bank Storage(-1)} + (\text{second Bank Storage Coefficient} \times (\text{Storage} - \text{Storage(-1)}))$$

The Change in Bank Storage is calculated using the following equation regardless of which method was used to compute Bank Storage.

$$\text{Change in Bank Storage} = \text{Bank Storage} - \text{Bank Storage(-1)}$$

where in the above equations:

Bank Storage = the volume of water stored in the banks of the Reservoir at the current timestep

Bank Storage(-1) = the volume of the water stored in the banks of the Reservoir at the previous timestep.

Storage = the volume of water in the Reservoir at the current timestep

Storage(-1) = the volume of water in the Reservoir at the previous timestep

23.1.26.4 Average Stage Change

The Average Stage Change method calculates the Bank Storage and Change in Bank Storage based on the flow from storage. The flow from storage is a function of the average stage change over a user defined number of timesteps.

Slots Specific to this Method

☛ Ave Stage Change Coeffs

Type: Table Slot
Units: Area per Time and Flow
Description: coefficient describing flow for a given change in pool elevation and a constant representing flow from bank storage
Information:
I/O: Required input
Links: Not linkable

☛ Bank Storage

Type: Series Slot
Units: Volume
Description: volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

☛ Change in Bank Storage

Type: Series Slot
Units: Volume
Description: change in volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

☛ TIMESTEPS TO AVERAGE

Type: Table Slot
Units: NO UNITS
Description: number of timesteps used to calculate average pool elevation.
Information:
I/O: Required input
Links: Not linkable

The average stage change is calculated using the following equation:

$$\text{average Pool Elevation} = \frac{\text{Pool Elevation} - \text{Pool Elevation (-Timesteps to Average)}}{\text{Timesteps to Average}}$$

The change in flow to bank storage is calculated using the following equation:

$$\text{Flow to banks} = \text{Average Stage Change Bank Storage Coefficient} \times \text{Average Pool Elevation} + \text{Average Stage Change Bank Storage Constant}$$

The flow is converted to a volume by multiplying the value by the current timestep. The Change in Bank Storage is calculated using the following equation:

$$\text{Change in Bank Storage} = \text{Bank Storage} - \text{Bank Storage(-1)}$$

23.1.27 Diversion from Reservoir

The Diversion from Reservoir user methods are applicable when a reservoir is linked to a diverting object (e.g. AggDiversionSite, AggDistributionCanal, or Diversion Object). These methods simply create the slots which must be linked (by the user) to slots on the diverting object.

23.1.27.1 None

This is the default for the Diversion from Reservoir category. It is used when the reservoir is not linked to a diverting object. If the reservoir is linked to a diverting object and this method is selected, the object will not solve correctly. There are no slots specifically associated with this method.

23.1.27.2 Available Flow Based Diversion

This method must be selected when a reservoir is linked to either an AggDiversionSite, AggDistributionCanal, or a Diversion Object that is using the Available For Diversion Linked method. Selecting this method allows the Available for Diversion slot to be available for linking. The AggDiversionSite, AggDistributionCanal, and Diversion objects contain more information about diverting water from a reservoir.

SLOTS SPECIFIC TO THIS METHOD

☛ **AVAILABLE FOR DIVERSION**

Type: Series
Units: FLOW
Description: represents the amount of water that may be diverted from the reservoir
Information:
I/O: Optional; can be input by the user or determined by **RiverWare**™.
Links: Should be linked to the Available for Diversion slot on AggDiversionSite or Diversion object, or the Incoming Available Water slot on a Water User.

Available for Diversion can either be input by the user or calculated by the reservoir. If it is not input it is set as the previous Storage divided by the timestep length. The value is limited to not be negative.

No other calculations are performed if this method is selected.

23.1.27.3 Head Based Diversion

This method may be selected when a reservoir is linked to a Diversion Object. Selecting this method allows the Previous Pool Elevation slot to be available for linking. The Diversion Object contains more information about diverting water from a Reservoir.

Slots Specific to this Method

☛ Previous Pool Elevation

Type: Series
Units: Length
Description: Pool Elevation value for the previous timestep
Information:
I/O: Output only
Links: Should be linked to the Diversion Intake Elevation slot on the Diversion Object.

23.1.28 Diversion Power

The methods in this category calculate power generated on the diversion from the reservoir. Selecting a method other than None in this category will make the Diversion Tailwater and Diversion Power Bypass categories available. If a method other than None is selected for Diversion Power, then a method other than None must be selected for Diversion Tailwater.

23.1.28.1 None

This is the default method for the Diversion Power category. No calculations are performed in this method, and there are no slots specifically associated with this method.

23.1.28.2 Diversion Power Efficiency Curve

The Diversion Power Efficiency Curve method is similar to the Plant Efficiency Curve method in the Power category with the exception that the method does not allow **Diversion Energy** to input or set by rules (nor can **Diversion Energy** be set with the Best Efficiency or Max Capacity flags). **Diversion Energy** and **Diversion Power** are only calculated as outputs. The method calculates **Diversion Power** by a 3-D interpolation of the **Diversion Power Table** using the current, average **Diversion Operating Head** and **Diversion Turbine Flow**. The **Diversion Power Coefficient** is calculated as **Diversion Power** divided by **Diversion Turbine Flow**. Alternatively, the user can input **Diversion Power Coefficient**, and then **Diversion Power** is calculated directly as the **Diversion Power Coefficient** multiplied by the **Diversion Turbine Flow**.

Slots Specific to this Method

☛ **Diversion Power Table**

Type: Table Slot
Units: Length vs Flow vs Power
Description: 3-D table representing the power characteristics of the diversion power plant, used to calculate power using interpolation
Information: Data must be entered into the table in increasing, blocks of the same Diversion Operating Head. For every block of the same Diversion Operating Head in column 1, Diversion Turbine Flow should be listed in increasing order in column 2 and the corresponding Diversion Power in column 3. The first row for each Diversion Operating Head must be for zero Diversion Turbine Flow and zero Diversion Power.
I/O: Required input
Links: Not linkable

Div Head	Turbine Flow	Div Power
30	0	0
30	100	100
30	200	175
40	0	0
40	100	125
40	220	195
50	0	0
50	110	147
50	250	205

Diversion Max Turbine Table

Type: Table Slot
Units: Length vs Flow
Description: The maximum Diversion Turbine Flow as a function of Diversion Operating Head
Information: RiverWare automatically populates this table at the start of the run using the Diversion Power Table. The first column contains the Diversion Operating Head values from the Diversion Power Table, one row for each unique Diversion Operating Head in increasing order. The second column contains the maximum Diversion Turbine Flow value for each Diversion Operating Head.
I/O: Output only
Links: Not Linkable

Diversion Power Cap Fraction

Type: Series Slot
Units: Fraction
Description: This is the percentage of full capacity of the turbine units in the diversion power plant. For example, if only half of the turbine are operational (and they are all the same), this value would be 0.5.
Information: This must be a number between 0 and 1 (inclusive). If not input or set by rules, this slot is automatically set to 1.
I/O: Optional input, if not, value is set to 1
Links: Not linkable

Diversion Operating Head

Type: Series Slot

Units: LENGTH
Description: The difference between the average Pool Elevation and the Diversion Tailwater Elevation
Information:
I/O: Output only
Links: Not usually linked

DIVERSION TURBINE FLOW

Type: Series Slot
Units: FLOW
Description: The diversion flow that passes through the turbines to generate power
Information: If the slot is not input or set by rules, then it is calculated as the difference between Diversion and Diversion Power Bypass if Diversion Power Bypass is input or set by rules. If neither Diversion Turbine Flow nor Diversion Power Bypass is input or set by rules, then Diversion Turbine Flow is calculated as the lesser of Diversion and the calculated maximum diversion turbine flow based on the Diversion Max Turbine Table and the current Diversion Operating Head. It is not permissible to have both Diversion Turbine Flow and Diversion Power Bypass as input or set by rules.
I/O: Optional input or output
Links: Not linkable

DIVERSION POWER

Type: Series Slot
Units: POWER
Description: The power generated from flow through the reservoir diversion
Information: If Diversion Power Coefficient is not input or set by rules, Diversion Power is calculated using a 3-D interpolation on the Diversion Power Table given the current, average Diversion Operating Head and the current Diversion Turbine Flow, scaled by the Diversion Power Cap Fraction. If Diversion Power Coefficient is input or set by rules, Diversion Power is calculated as the Diversion Power Coefficient multiplied by the Diversion Turbine Flow.
I/O: Output only
Links: Linkable

DIVERSION ENERGY

Type: Series Slot
Units: ENERGY
Description: The energy generated from flow through the reservoir diversion
Information: Calculated as the Diversion Power multiplied by the timestep length
I/O: Output only

Links: Linkable

☛ Diversion Power Coefficient

Type: Series Slot
Units: Power per Flow
Description: The power generation per unit of flow through the turbines on the reservoir diversion
Information: If this slot is input or set by rules, it is used directly to calculate Diversion Power. If it is not input or set by rules, then it is calculated as Diversion Power divided by Diversion Turbine Flow. If either Diversion Power, Diversion Turbine Flow or Diversion Power Cap Fraction is zero, then this slot will be zero.
I/O: Optional input or output
Links: Not usually linked

At the start of the run, the **Diversion Max Turbine Table** slot is populated using the **Diversion Power Table**. The first column is populated with each unique **Diversion Operating Head** value from the **Diversion Power Table**, in ascending order. The second column is populated with the corresponding maximum diversion turbine flow value.

When the method executes, **Diversion** will already be known. The method calls the selected Diversion Tailwater method to calculate the **Diversion Tailwater Elevation**. If the default method, **None**, is selected for the **Diversion Tailwater** category, the run will abort and an error message will be issued.

The method then calculates the **Diversion Operating Head**:

$$\text{Diversion Operating Head}[t] = \frac{\text{Pool Elevation}[t-1] + \text{Pool Elevation}[t]}{2} - \text{Diversion Tailwater Elevation}[t]$$

The method calculates the *maxDiversionTurbineFlow* by interpolating the **Diversion Max Turbine Table** slot using the **Diversion Operating Head**. The value is scaled by the **Diversion Power Cap Fraction**.

If **Diversion Turbine Flow** is specified (input or set by rules) it is checked against the *maxDiversionTurbineFlow*, and if the specified value exceeds the max, the run will abort with an error message. Otherwise a temporary turbine flow is calculated.

$$\text{tempDiversionTurbine} = Min(\text{Diversion} - \text{Diversion Power Bypass}, maxDiversionTurbineFlow)$$

If **Diversion Power Bypass** is not input or set by rules, or if the **Diversion Power Bypass** method is **None**, the **Diversion Power Bypass** will have defaulted to zero at this point.

If the combined temporary turbine flow plus the current **Diversion Power Bypass** is less than the (total) **Diversion**, then the method then calls the selected Diversion Power Bypass method to increase the **Diversion Power Bypass** to make up the difference. If it is not possible for the turbine flow plus the bypass to equal the total **Diversion**, either due to the values being specified (input or rules) or due to max capacity limits, then the run will abort with an error message.

The method then sets the Diversion Turbine Flow slot:

$$\text{Diversion Turbine Flow} = \text{tempDiversionTurbine}$$

If **Diversion Power Coefficient** is specified (input or rules), it is used to calculate **Diversion Power** directly:

$$\text{Diversion Power} = \text{Diversion Turbine Flow} \times \text{Diversion Power Coefficient}$$

Otherwise, **Diversion Power** is calculated by a 3-D interpolation on the **Diversion Power Table** using **Diversion Operating Head** and **Diversion Turbine Flow**, and **Diversion Power Coefficient** is calculated as:

$$\text{Diversion Power Coefficient} = \frac{\text{Diversion Power}}{\text{Diversion Turbine Flow}}$$

Diversion Energy is then calculated as **Diversion Power** multiplied by the timestep length.

Notes on Diversion Power Cap Fraction

If the **Diversion Power Cap Fraction** is input by the user, it is necessary for the **Diversion Power Table** to be scaled back to account for the operating points when the turbines are operating at less than 100%. To do this, **Diversion Turbine Flow** is divided by the **Diversion Power Cap Fraction**. This point is then found in the **Diversion Power Curve** for the current **Diversion Operating Head**, and the power is found using 3-D interpolation. Finally the power is multiplied by the **Diversion Power Cap Fraction** to get the actual **Diversion Power** produced for the current timestep.

23.1.29 Diversion Tailwater

The methods in this category calculate the elevation of the tailwater on the diversion from a reservoir. This category is dependent on the selection of a method other than the default method, None, in the Diversion Power category. If a method other than the default is selected for Diversion Power, then a method other than the default, None, must be selected for Diversion Tailwater.

23.1.29.1 None

This is the default method for the Diversion Tailwater category. No calculations are performed in this method, and there are no slots specifically associated with this method.

23.1.29.2 Diversion Base Value Plus Lookup

The Diversion Base Value Plus Lookup method computes the **Diversion Tailwater Elevation** by added the average **Diversion Tailwater Base Value** (over the timestep) to a function of **Diversion** defined in the **Diversion Tailwater Table** slot. This method is similar to the Base Value Plus Lookup Table method in the Tailwater category but uses the **Diversion** and **Diversion Tailwater Base Value** slots instead of **Outflow** and **Tailwater Base Value**. The **Diversion Tailwater Base Value** may be input by the user or linked to another slot, such as the **Pool Elevation** of another Reservoir. If the **Tailwater Base Value** is neither input nor linked, it is automatically set to zero.

Slots Specific to this Method

☛ **Diversion Tailwater Table**

Type: Table Slot
Units: Flow vs Length
Description: This slot defines the relationship between Diversion and the Diversion Tailwater Elevation; Diversion vs either the diversion tailwater elevation or the tailwater elevation increment
Information: If the Diversion Tailwater Base Value is non-zero, the Diversion Tailwater Table gives values of incremental increase in Tailwater Elevation over th base value. Otherwise, the table gives the Diversion Tailwater Elevation values. The first row of the table should be for a Diversion flow of zero.
I/O: Required input
Links: Not linkable

☛ **Diversion Tailwater Base Value**

Type: Series Slot
Units: Length
Description: the base elevation of the diversion tailwater, such as a downstream stage

Information: If the slot is not input or linked, it defaults to 0.
I/O: Optional, can be input or linked
Links: Linkable

DIVERSION TAILWATER ELEVATION

Type: Series Slot
Units: LENGTH
Description: the water surface elevation of the tailwater from the reservoir diversion
Information: This slot is used to compute Diversion Operating Head in Diversion Power calculations
I/O: Output only
Links: Not linkable

When this method is executed, the **Diversion** value will already be known. If the **Diversion Tailwater Base Value** is neither linked, input nor set by rules, then it will default to zero.

The following steps are performed to calculate **Diversion Tailwater Elevation**.

1. *TWTemp* is obtained from a table interpolation on the **Diversion Tailwater Table** using **Diversion**.

2. If both **Diversion Tailwater Base Value[t]** and **Diversion Tailwater Base Value[t-1]** are known, then the **Diversion Tailwater Elevation** is calculated as:

$$\text{Diversion Tailwater Elevation}[t] = \frac{\text{Diversion Tailwater Base Value}[t-1] + \text{Diversion Tailwater Base Value}[t]}{2} + TWTemp$$

3. If **Diversion Tailwater Base Value[t]** is known, but **Diversion Tailwater Base Value[t-1]** is not known, then the **Diversion Tailwater Elevation** is calculated as:

$$\text{Diversion Tailwater Elevation}[t] = \text{Diversion Tailwater Base Value}[t] + TWTemp$$

4. If **Diversion Tailwater Base Value[t-1]** is known, but **Diversion Tailwater Base Value[t]** is not known, then the **Diversion Tailwater Elevation** is calculated as:

$$\text{Diversion Tailwater Elevation}[t] = \text{Diversion Tailwater Base Value}[t-1] + TWTemp$$

5. If neither **Diversion Tailwater Base Value[t]** nor **Diversion Tailwater Base Value[t-1]** are known but **Diversion Tailwater Elevation[t-1]** is known, the current timestep's **Diversion Tailwater Elevation** is set equal to **Diversion Tailwater Elevation[t-1]**.

6. If neither **Diversion Tailwater Base Value[t]**, **Diversion Tailwater Base Value[t-1]**, nor **Diversion Tailwater Elevation[t-1]** are known, or if **Diversion** is not known, the method will exit and wait for more information.

23.1.30 Diversion Power Bypass

The methods in this category calculate the portion of the diversion from a reservoir that does not pass through the turbines but rather through a bypass structure. This category is dependent on the selection of a method other than the default method, None, in the Diversion Power category.

23.1.30.1 None

This is the default method for the Diversion Power Bypass category. No calculations are performed in this method, and there are no slots specifically associated with this method. If this method is selected, it is assumed that all **Diversion** flow passes through the turbines.

23.1.30.2 Bypass Capacity Table

This method sets **Diversion Power Bypass** to the difference between **Diversion** and **Diversion Turbine Flow** if it is not input or set by rules, and it checks that the **Diversion Power Bypass** does not exceed the maximum based on the **Diversion Power Bypass Table**. This functions similarly to the Regulated method in the Spill category.

Slots Specific to this Method

☛ **Diversion Power Bypass Table**

Type: Table Slot
Units: Length vs Flow
Description: Pool Elevation vs. the corresponding maximum diversion power bypass values
Information:
I/O: Required input
Links: Not linkable

☛ **Diversion Power Bypass**

Type: Series
Units: Flow
Description: Diversion flow that does not pass through power turbines
Information: If not input or set by rules, Diversion Power Bypass will be set equal to the difference between Diversion and Diversion Turbine Flow.
I/O: Optional input or output
Links: Linkable

At the beginning of the run, if **Diversion Power Bypass** is not specified (input or rules), it is initially set to a default value of zero.

On each timestep, the method first checks if both **Diversion Power Bypass** and **Diversion Turbine Flow** are input or set by rules. If so, the run will abort with an error message. It necessary to leave at least one of these slots as a free variable.

The method then calculates max diversion power bypass by performing a table interpolation on the **Diversion Power Bypass Table** using the average **Pool Elevation** from the end of the current timestep and end of the previous timestep.

If **Diversion Power Bypass** is input or set by a rule, then the value is checked against the max diversion power bypass, and if it exceeds the max, the run will abort with an error message.

If **Diversion Power Bypass** is not input or set by a rule, then it is calculated as:

$$\text{Diversion Power Bypass} = \text{Diversion} - \text{Diversion Turbine Flow}$$

The calculated **Diversion Power Bypass** value is checked against the max diversion power bypass, and if it exceeds the max, the run will abort with an error message.

23.1.31 Seepage

The Seepage methods are used to calculate the amount of water lost through the face of the dam. The volume of seepage computed during the execution of these methods affects the mass balance of the Reservoir.

23.1.31.1 None

None is the default for the Seepage category. It is used when the user does not want to calculate the flow of water through the face of the dam.

23.1.31.2 Input Seepage

The Input Seepage method is used when it is desired to have the seepage slot as input or set by a rule.

SLOTS ADDED BY THIS METHOD:

☛ **SEEPAGE**

Type: SeriesSlot
Units: FLOW
Description: flow of water through the dam face
Information: Seepage is not included in the Outflow of the reservoir and will need to be linked separately if the water does in fact go downstream.
I/O: Input only
Links: Linkable

23.1.31.3 Linear Seepage

The Linear Seepage method calculates the seepage from the face of the dam. This calculation is based on the Pool Elevation of the Reservoir and specified coefficients.

SLOTS ADDED BY THIS METHOD:

☛ **SEEPAGE**

Type: SeriesSlot
Units: FLOW
Description: flow of water through the dam face
Information: Seepage is not included in the Outflow of the reservoir and will need to be linked separately if the water does in fact go downstream.
I/O: Output only
Links: Linkable

☛ **SEEPAGE COEFFICIENTS**

Type: TableSlot
Units: LENGTH, AREA PER TIME, FLOW
Description: coefficients in the linear equation for seepage
Information: The first coefficient (column zero) is the base elevation of the dam. The second coefficient is the slope of the linear equation for seepage. The third coefficient is the intercept of the linear equation for seepage.
I/O: Required Input
Links: NA

The calculation for Seepage in this method is fairly straightforward. A linear model is used. The coefficient are user inputs. The following equation is used to compute Seepage:

$$\text{Seepage} = (\text{Pool Elevation} - \text{first Seepage Coefficient}) \times \text{second Seepage Coefficient} + \text{third Seepage Coefficient}$$

23.1.31.4 Single Seepage Value

The Single Seepage Value method sets the seepage from the face of the dam equal to a scalar value.

SLOTS ADDED BY THIS METHOD:

☛ **SEEPAGE**

Type: Series
Units: FLOW
Description: Flow of water through the dam face
Information: Seepage is not included in the Outflow of the reservoir and will need to be linked separately if the water does in fact go downstream.
I/O: Optional Input
Links: Linkable

☛ **SINGLE SEEPAGE VALUE**

Type: Scalar
Units: FLOW
Description: seepage value to be applied to each timestep
Information:
I/O: Required Input
Links: Not Linkable

This method is executed at the beginning of the run. For each timestep from the initial timestep through the end of the run (plus post run dispatching timesteps too), if the **Seepage** is not input, the **Seepage** is set equal to the **Single Seepage Value**.

The method will issue an error if there is not a valid value in the **Single Seepage Value** slot. Note, this structure allows some flexibility. Seepage can be input/rules when necessary but will use the scalar value when not input.

23.1.31.5 Linked Seepage

This method is intended to be used when linking a Reservoir object with a Groundwater Storage object that uses the **Head Based Boundary Condition** method in the **Solution Type** category HERE (Objects.pdf, Section 14.1.1.3).

SLOTS ADDED BY THIS METHOD:

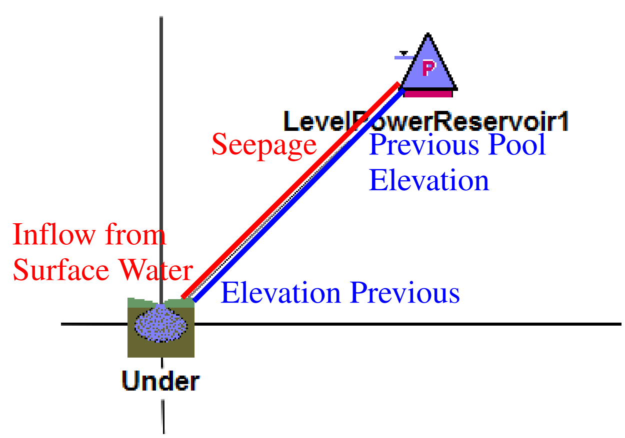
SEEPAGE

Type: Series
Units: FLOW
Description: Flow of water out of the reservoir, often into groundwater
Information: A positive value is flow out of the reservoir.
I/O: Output only if linked to a Groundwater object (typical); otherwise required input
Links: Must be linked, typically to **Inflow from Surface Water** on Groundwater object

PREVIOUS POOL ELEVATION

Type: Series
Units: LENGTH
Description: Pool Elevation at the end of the previous timestep
Information:
I/O: Output only
Links: Typically linked to the **Elevation Previous** slot on a Groundwater object

This method does not do any calculations; it just adds the appropriate slots. The Reservoir does provide the **Previous Pool Elevation** which is then linked to the Groundwater object **Elevation Previous**. The Groundwater object computes the **Inflow from Surface Water**, which is linked back to the **Seepage** slot on the reservoir. The **Seepage** is used in the Reservoir mass balance.



Click **HERE (Objects.pdf, Section 14.1.1.3)** for more information about how **Seepage** is calculated as **Inflow from Surface Water** on the linked Groundwater object.

23.1.32 Operating Levels

This category enables the user to specify operating levels for the reservoir. Operating levels serve as a normalizing metric for reservoir contents. This metric is used by reservoir-balancing algorithms to determine the relative "fullness" of reservoirs. On individual reservoirs, it also serves to identify the elevations that correspond to pool boundaries, such as the top of the conservation pool or the top of the flood pool.

23.1.32.1 None

This is the default method; no slots are instantiated and no calculations are performed.

23.1.32.2 Conservation Pools

This method allows the user to specify that there is a conservation pool for this reservoir.

SLOTS SPECIFIC TO THIS METHOD

☛ **OPERATING LEVEL TABLE**

Type: PeriodicSlot
Units: TIME VS LENGTH AT OPERATING LEVELS
Description: table describing the seasonal variation of elevation in a reservoir at each of the user-designated operating levels.
Information: number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep.
I/O: Required Input
Links: Not Linkable

☛ **OPERATING LEVEL**

Type: SeriesSlot
Units: NONE
Description: The computed operating level
Information: This slot is computed using the pool elevation and the Operating Level Table
I/O: Output only
Links: Not Linkable

☛ Operating Level Storage Table

Type: PeriodicSlot
Units: Time vs Volume at Operating Levels
Description: table describing the seasonal variation of storage in a reservoir at each of the user-designated operating levels.
Information: Each column represents the time-varying storage for a particular Operating Level. The integer value of the Operating Level is in the header of each column. This table is generated from the Operating Level Table and has the same number of rows and columns. The values in the table are storage values (looked up from the elevation volume table) whereas the values in the Operating Level Table are elevations. This slot is computed at run-time so it is read-only to the user. All changes should be made in the Operating Level Table.
I/O: Output Only
Links: Not Linkable

☛ Top of Conservation Pool

Type: ScalarSlot
Units: NONE
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the conservation pool.
Information:
I/O: Input
Links: Not Linkable

☛ Bottom of Conservation Pool

Type: ScalarSlot
Units: NONE
Description: Operating level (as defined in Operating Level Table) corresponding to the bottom of the conservation pool.
Information: Used by some conservation pool operations algorithms, along with the Top of Conservation Pool slot, to identify the volume in the conservation pool.
I/O: Input Only
Links: Not Linkable

☛ Conservation Pool Initial Empty Space

Type: SeriesSlot
Units: None
Description: The inflow required to fill the conservation pool at the beginning of timestep, based on the ending storage at the prior timestep, taking into account evaporation, precipitation, etc.

Information: This slot is computed at the beginning of the timestep; evaporation rates and other such factors that are not already defined at the beginning of timestep will not be taken into account in this computation. This slot is used by the water rights allocation solution algorithm to compute physical constraints and by storage accounts to compute allocation requests.

I/O: Output only

Links: Not Linkable

☛ Conservation Pool Storage

Type: Series

Units: Volume

Description: This is the computed volume of water in the conservation pool.

Information: This value is always non-negative.

I/O: Output only

Links: Not Linkable

☛ Conservation Pool Full Storage

Type: Series

Units: Volume

Description: This is the possible volume of water that could be stored in the Conservation pool. It is computed as the Storage at the top of the conservation pool minus the storage at the bottom of the conservation pool

Information:

I/O: Output Only

Links: Not Linkable

☛ Conservation Pool Storage Fraction

Type: Periodic Slot

Units: Fraction

Description: The values in the periodic slot represent the percentage of the conservation pool storage at each level (column) in the Operating Level Table.

Information: It has identical dimension including dates and levels as the Operating Level Table. This table will be populated at beginning of run. The Operating Level Table will be its “source” slot.

I/O: Output only

Links: Not Linkable

At the beginning of run, the Conservation Pool Storage Fraction is populated as follows: For each date (row) and each level, n, (column), the equation to compute the fraction:

$$\text{Conservation Pool Storage Fraction}[date,n]=$$

$$\frac{\text{Op Level Storage Table}[date, n] - \text{Op Level Storage Table}[date, \text{Bottom of Cons Pool}]}{\text{Op Level Storage Table}[date, \text{Top of Cons Pool}] - \text{Op Level Storage Table}[date, \text{Bottom of Cons Pool}]}$$

Note, the Conservation Pool Storage Fraction is limited to be always between 0 and 1 (0% to 100%).

At the end of each dispatch method, the Operating Level series slot is computed by looking up the pool elevation and date on the Operating Level Table.

Next, **Conservation Pool Full Storage** is calculated as follows.

$$\text{Conservation Pool Full Storage}[t] = \text{Operating Level Storage Table}[t, \text{Top of Conservation Pool}] - \text{Operating Level Storage Table}[t, \text{Bottom of Conservation Pool}]$$

The **Conservation Pool Storage** is computed as:

If the Operating Level is greater than the Top of the Conservation Pool,

$$\text{Conservation Pool Storage}[t] = \text{Full Conservation Pool Storage}[t]$$

else if the Operating Level is less than the Bottom of the Conservation Pool,

$$\text{Conservation Pool Storage}[t] = 0$$

else

$$\text{Conservation Pool Storage}[t] = \text{Storage}[t] - \text{Operating Level Storage Table}[t, \text{Bottom of Conservation Pool}]$$

23.1.32.3 Conservation and Flood Pools

This method allows the user to specify that there is a conservation and a flood pool for this reservoir.

Slots Specific to this Method

This method is an extension of the Conservation Pool method, and selecting this method causes all the slots for Conservation Pool to become available, along with the following:

Top of Flood Pool

Type: ScalarSlot
Units: No Units
Description: Operating level (as defined in Operating Level Table) corresponding to the top of flood pool.
Information:
I/O: Required Input
Links: Not Linkable

☛ Flood Pool Storage

Type: Series
Units: VOLUME
Description: This is the computed volume of water in the flood pool.
Information: This value is always non-negative.
I/O: Output Only
Links: Not Linkable

☛ Flood Pool Full Storage

Type: Series
Units: VOLUME
Description: This is the possible volume of water that could be stored in the Flood pool. It is computed as the Storage at the top of the flood pool minus the storage at the top of the conservation pool.
Information:
Links: Not Linkable

☛ Flood Pool Storage Fraction

Type: Periodic Slot
Units: FRACTION
Description: The values in the periodic slot represent the percentage of the flood pool storage at each level (column) in the Operating Level Table.
Information: It has identical dimension including dates and levels as the Operating Level Table. This table will be populated at beginning of run. The Operating Level Table will be its “source” slot.
I/O: Output only
Links: Not Linkable

At the beginning of run, the Conservation Pool Storage Fraction is populated as described above. Then, the Flood Pool Storage Fraction is populated as follows: For each date (row) and each level, n, (column), the equation to compute the fraction is:

$$\text{Flood Pool Storage Fraction}[date,n] = \frac{\text{Op Level Storage Table}[t,n] - \text{Op Level Storage Table}[t, \text{Top of Cons Pool}]}{\text{Op Level Storage Table}[t, \text{Top of Flood Pool}] - \text{Op Level Storage Table}[t, \text{Top of Cons Pool}]}$$

Note, the Flood Pool Storage Fraction is limited to be always greater than 0. But, it can be larger than 1 (100%). For levels above the flood pool, the percentage will be greater than 100%.

At the end of each dispatch method, the Operating Level series slot is computed by looking up the pool elevation and date on the Operating Level Table. Next, all slots associated with the Conservation Pool are computed and set as described above. Then, **Flood Pool Full Storage** is calculated as follows:

$$\text{Flood Pool Full Storage}[t] = \text{Operating Level Storage Table}[t, \text{Top of Flood Pool}] - \text{Operating Level Storage Table}[t, \text{Top of Conservation Pool}]$$

The **Flood Pool Storage** is computed as:

If the Operating Level is less than the Top of the Conservation Pool,

$$\text{Flood Pool Storage}[t] = 0$$

else

$$\text{Flood Pool Storage}[t] = \text{Storage}[t] - \text{Operating Level Storage Table}[t, \text{Top of Conservation Pool}]$$

Note, the **Flood Pool Storage** may be larger than the **Flood Pool Full Storage**. This indicates the reservoir is above the flood pool and is surcharging.

23.1.33 Conditional Operating Levels

This category provides methods that allows the user to use alternative operating level tables based on conditions in the run.

23.1.33.1 None

This is the default method; no slots are instantiated and no calculations are performed. The original Operating Level Table is used for all computations.

23.1.33.2 Sum Inflows over Interval

This method allows an alternative operating level table (i.e. a guide curve) to be used starting on a certain date if a certain combination of flows are high enough for a specified time range.

For example, if there has been a total of 200,000 acre-feet of total inflows into a specific reservoir during the months of March, April, and May, then on June 15th, the method would switch the reservoir operations to follow an alternative table. On October 15th, the reset date, the reservoir will once again use the original Operating Level Table.

Slots Specific to this Method

☛ Operating Level 2 Table

Type: Periodic
Units: Time vs Length at Operating Levels
Description: This is the alternative operating level table that is used when indicated by hydrologic conditions. This table describes the seasonal variation of elevation in a reservoir at each of the user-designated operating levels.
Information: Number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep. This table should have the same dimensions (rows and columns) as the Operating Level Table.
I/O: Required Input
Links: Not Linkable

☛ Operating Level 2 Trigger Volume

Type: Scalar
Units: Volume

Description: The volume of inflows (**Inflow Sum** slot) between the criteria start and finish (specified on the **Operating Level 2 Dates** slot) that causes the reservoir to use the **Operating Level 2 Table** rather than the original **Operating Level Table.**

I/O: Required Input or an error will be issued at the start of run.

Links: NA

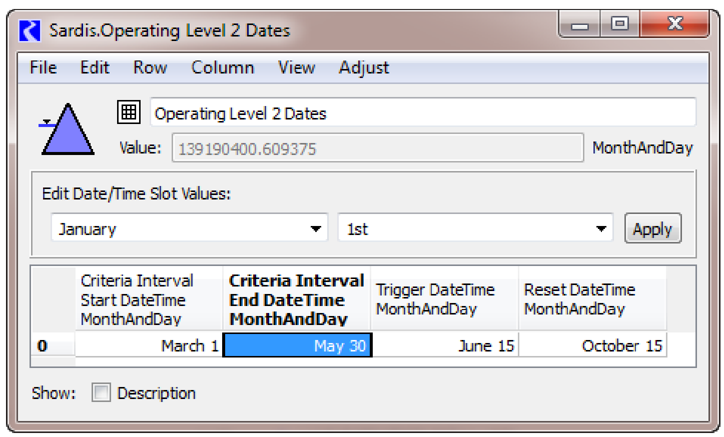
☛ Operating Level 2 Dates

Type: Table

Units: DateTime

Description: This table has 4 columns. The first and second columns are the “Criteria Interval Start DateTime” and “Criteria Interval End DateTime” between which the flow volume is summed and compared to the **Operating Level 2 Trigger Volume**. The third column is the “Trigger DateTime” at which the reservoir will check the conditions and possibly use the **Operating Level 2 Table** rather than the original **Operating Level Table.** The fourth column is the “Reset DateTime” at which the reservoir will use the **Operating Level Table** again.

Information: The units for this slot are DateTime which can be an absolute or partially specified datetime. The default user units are “MonthAndDay”. Partially specified datetimes are converted to fully specified datetimes using information from the current timestep to fill in the missing pieces of the partially specified timestep. Thus, if the datetime is partially specified, it must be able to evaluate to a timestep in the model or an error will be issued.



I/O: Required input or an error will be issued at the start of run.

Links: NA

☛ Operating Level Storage 2 Table

Type: Periodic

Units: Storage

Description: This represents the Storage associated with the elevations in the Operating Level 2 Table.

Information: This slot is created at the beginning of run. The Operating Level 2 Table is its "source" slot.
I/O: Output Only
Links: Not Linkable.

☛ Conservation Pool Storage Fraction 2

Type: Periodic Slot
Units: Fraction
Description: The values in the periodic slot represent the percentage of the conservation pool storage at each level (column) in the Operating Level Storage 2 Table.
Information: It has identical dimension including dates and levels as the Operating Level Storage 2 Table. This table will be populated at beginning of run. The Operating Level 2 Table will be its "source" slot.
I/O: Output only
Links: Not Linkable

☛ Flood Pool Storage Fraction 2

Type: Periodic Slot
Units: Fraction
Description: The values in the periodic slot represent the percentage of the flood pool storage at each level (column) in the Operating Level 2 Table.
Information: It has identical dimension including dates and levels as the Operating Level Storage 2 Table. This table will be populated at beginning of run. The Operating Level 2 Table will be its "source" slot.
I/O: Output only
Links: Not Linkable

Method Details:

This method category will be dependent on the selection of the **Conservation and Flood Pools** or **Conservation Pool** method in the **Operating Levels** method category.

At the beginning of the run, the Operating Level Storage 2 Table will be populated by looking up the elevation values in the Operating Level 2 Table on the Elevation Volume Table to get the storage associated with each level. Next the Conservation Pool Storage Fraction 2 and Flood Pool Storage Fraction 2 slots will be populated as follows:

For each date (row) and each level, n, (column), the equation to compute the fraction:

$$\text{Conservation Pool Storage Fraction 2}[date,n] =$$

$$\frac{\text{Op Level Storage 2 Table}[t, n] - \text{Op Level Storage 2 Table}[t, \text{Bottom of Cons Pool}]}{\text{Op Level Storage 2 Table}[t, \text{Top of Cons Pool}] - \text{Op Level Storage 2 Table}[t, \text{Bottom of Cons Pool}]}$$

Note, the **Conservation Pool Storage Fraction 2** is limited to be always between 0 and 1 (0% to 100%).

For each date (row) and each level, n, (column), the equation to compute the fraction:

$$\text{Flood Pool Storage Fraction}[date, n] = \frac{\text{Op Level Storage 2 Table}[t, n] - \text{Op Level Storage 2 Table}[t, \text{Top of Cons Pool}]}{\text{Op Level Storage 2 Table}[t, \text{Top of Flood Pool}] - \text{Op Level Storage 2 Table}[t, \text{Top of Cons Pool}]}$$

Note, the Flood Pool Storage Fraction 2 is not limited to be between 0 and 1 (0% to 100%). For levels above the flood pool, the percentage will be greater than 100%.

Then, at the beginning of each timestep, the Sum Inflows over Interval method will check to see if the controller is on the “Trigger DateTime”. If so, the Inflow Sum slot will be summed (as a volume) over the criteria interval. If the sum is equal to or greater than the Operating Level 2 Trigger Volume, the reservoir will use the Operating Level 2 Table and Operating Level Storage 2 Table in all computations (until reset).

If the current timestep is a “Reset DateTime”, then the reservoir will again use the original Operating Level Table(s).

If the current timestep is neither a “Trigger DateTime” or a “Reset DateTime”, then the reservoir will reference the table used on the previous timestep. That is, it will not modify the table used but continue to use whichever table is in effect.

23.1.34 Startup

This category depends on selecting the Unit Power Table method, and describes how the monetary cost associated with starting up or shutting down a unit (turbine) will be modeled. There are two methods in this category, one which does not model these costs (effectively assigning them a value of 0) and one which uses a table describing the combined costs for starting up and shutting down a unit.

23.1.34.1 No Method (default)

This is the default, do-nothing method.

23.1.34.2 Unit Lumped Cost Method

For each Unit, this method lumps the cost of startup and shutdown into one value.

SLOTS ADDED BY THIS METHOD

Note, many of these slots have column or row dimensions based on the number of units. The rows/columns of these slots are expanded at the beginning of the run to match the value in the Number of Units slot. When first configuring this method, the user must enter the Number of Units, then run the model (stepping through 1 timestep is enough) to grow the slots to the right dimensions.

The following slots are instantiated when this method is selected:

☛ **UNIT STARTUP COST TABLE**

Type: TableSlot
Units: VALUE ($)
Description: This table will indicate the cost of startup/shutdown of each unit.
Information: There will be one column for each unit and one row that represents the cost of startup/shutdown.
I/O: Required input
Links: NA

☛ **UNIT STARTUP COST**

Type: AggSeriesSlot
Units: VALUE ($)
Description: There is one column for each unit indicating the cost of startup/shutdown.
Information: In simulation, the value of Unit Startup Cost for each unit is the Unit Startup[u] * Unit Startup Cost Table [u].
I/O: Output only
Links: NA

☛ Unit Startup

Type: AggSeriesSlot
Units: No Units
Description: A value of 1 indicates that the unit starts up at that date; otherwise the value is 0, indicating that the unit does not start up at that date.
Information: There is one column for each unit.
I/O: Output only
Links: NA

☛ Unit Shutdown

Type: AggSeriesSlot
Units: No Units
Description: A value of 1 indicates that the unit shuts down at that date; otherwise the value is 0, indicating that the unit does not shut down at that date.
Information: There is one column for each unit.
I/O: Output only
Links: NA

☛ Number Of Units Startup

Type: SeriesSlot
Units: No Units
Description: The number of units which start up at a given date. This value is the sum over the columns of Unit Startup.
Information:
I/O: Output only
Links: NA

☛ Number Of Units Shutdown

Type: SeriesSlot
Units: No Units
Description: The number of units which shut down at a given date. This value is the sum over the columns of Unit Shutdown.
Information:
I/O: Output only
Links: NA

☛ Plant Startup Cost

Type: SeriesSlot
Units: Value ($)
Description: The total startup cost for the plant. This value is the sum over the columns of the Unit Startup Cost.
Information:

I/O: Output only

Links: Linkable, typically to a Thermal object's System Startup slot.

METHOD DETAILS

In Simulation, if the Unit Lumped Cost method is selected, startup and shutdown will be summarized as follows:

- Calculate Unit Startup[t,u] = max(Unit Is Generating[t,u] - Unit Is Generating[t-1,u], 0)
- Calculate Unit Shutdown[t,u] = max(Unit Is Generating[t-1,u] - Unit Is Generating[t,u], 0)
- Calculate Number Of Units Startup[t] = sum(Unit Startup[t])
- Calculate Number Of Unit Shutdown[t] = sum(Unit Shutdown[t])

Note, if the previous Unit Is Generating is not known, it is assumed that the unit is neither starting up or shutting down; Unit Startup and Unit Shutdown are set to zero. This may happens on the start timestep when the previous value is not known. Also, if the current Unit Is Generating is not valid, the method is exited without performing any computations or setting any slots.

This method will calculate the cost associated with startup/shutdown for each unit and the plant:

- Unit Startup Cost[t,u] = Unit Startup[t,u] * Unit Startup Cost Table[u]
- Plant Startup Cost[t] = Σ (Unit Startup Cost[t])

23.1.35 Head Loss

This category depends on the Unit Power Table method and contains methods for modeling additional head loss that occurs. This head loss may come from the configuration of the penstocks for bringing water to the turbines.

23.1.35.1 No Method (default)

In this method, there is no additional head loss to be used in the power calculation. In terms of penstock head loss, this method should be selected if the penstocks for the units are independent and the penstock losses are typically incorporated in the power data. Thus the power data is specified in terms of operating head.

23.1.35.2 Shared Penstock Head Loss method

In this method, there is additional head loss that results because units share a common penstock. The operating head losses in the penstock depend on the total turbine release and are shared for all units. The net head is calculated by subtracting penstock losses from the operating head. The unit data and power must be specified in terms of unit Net Heads instead of Operating Head.

Slots Added by this Method

The following slots are instantiated when this method is selected:

☛ **Shared Penstock Head Loss Table**

Type: TableSlot
Units: Flow vs Length
Description: This table shows head losses in a shared penstock as a function of total turbine release.
Information: The table has two columns: Turbine Release and Shared Penstock Loss.
I/O: Required Input
Links: NA

Method Details

In simulation, when either the Unit Power Table or Unit Power Table Release method is called, it has an estimate of operating head calculated as average PE minus average tailwater elevation. Within these methods, if the Shared Penstock Head Loss method is selected, the code will look up an estimate of total Turbine Release and compute the additional head loss and subtract it from the existing operating head before use in any other equation. An iterative solution is needed if Turbine Release is not known, i.e. if this method is called from the Unit Power Table Release method.

23.1.36 Cavitation

This category depends on selecting the Unit Power Table method and contains methods for dealing with the problem of cavitation on turbines. Cavitation is the sudden formation and collapse of low-pressure bubbles in liquids by means of mechanical forces and this process can cause damage to turbines under certain operating conditions.

23.1.36.1 No Method (default)

This is the default, do-nothing method.

23.1.36.2 Unit Head and Tailwater Based Regions

This method allows the user to specify the regions of operation in which cavitation does NOT occurs, so that these regions can be avoided. These regions can be dependent on both operating head and tailwater.

Slots added by this method

Note, many of these slots have column or row dimensions based on the number of units. The rows/columns of these slots are expanded at the beginning of the run to match the value in the Number of Units slot. When first configuring this method, the user must enter the Number of Units, then run the model (stepping through 1 timestep is enough) to grow the slots to the right dimensions.

The following slots are instantiated when this method is selected:

☛ **Unit Power Cavitation Table**

Type: TableSlot
Units: Length, Length, Power, and Power
Description: This table represents the region of operation that does not cause cavitations.
Information: The table will have one block per unit and four columns per block: head, tailwater, and minimum power to prevent cavitation, and maximum power to prevent cavitation. Interpolation of this table will be used to determine the feasibility for each flow - head combinations in the Unit Power Table. Some combinations may not be feasible at any tailwater, and these combinations should not be used in optimization or simulation. Others will have a minimum tailwater level for feasibility. For some units, tailwater may not affect cavitation. In these cases two rows should be used for each head: one with minimum tailwater and one with maximum tailwater.
I/O: Required Input
Links: NA

☛ **Unit Cavitation Optimization Tolerances**

Type: Table

Unit 1				Unit 2			
Tailwater (ft)	Operating Head (ft)	Min. Non-Cav. Power (MW)	Max. non-Cav. Power (MW)	Tailwater (ft)	Operating Head (ft)	Min. Non-Cav. Power (MW)	Max. non-Cav. Power (MW)
2067	100	1	12	2067	100	2	10
2067	200	1.1	12.5	2067	200	3	11
2067	300	1.4	12.6	2067	300	3.5	12
2116	100	2	12	2116	100	2	10
2116	200	2.1	12.3	2116	200	3	11
2116	300	2.5	12.8	2116	300	3.5	12

Units: FRACTION, FRACTION
Description: Tolerance used to adjust the cavitation region in Optimization.
Information: This slot is used to “shrink” the cavitation region in Optimization to avoid the possibility of optimal solutions that when run in the Rulebased Simulation, just barely dip into a cavitation zone. For example, if 0.01 is specfied, this translates to giving the optimization a 1% cushion to avoid the cavitation zone. There is one row for each unit.
I/O: Optional Input
Links: NA

METHOD DETAILS

In Simulation, the dispatch method will execute this method once head, tailwater elevation, and power are computed for each unit. This method will determine if the computed head, tailwater elevation, and power fall outside of the minimum and maximum power to prevent cavitation regions. If so, the method will issue an error but not stop the run. If the method is called and there is no valid Unit Power, then the method is exited without performing any computations.

23.1.37 Avoidance Zones

This category depends on selecting the Unit Power Table method and contains methods for modeling the existence of undesirable regions of operation for turbines. There are two methods in this category, one which does not model avoidance zones at all, and one which

23.1.37.1 No Method (default)

This the default, do-nothing method; avoidance zones are not considered.

23.1.37.2 Unit Head Based Avoidance Zones

This method allows the user to specify a table that defines the conditions in which the turbines should not be operated.

Slots added by this method

Note, many of these slots have column or row dimensions based on the number of units. The rows/columns of these slots are expanded at the beginning of the run to match the value in the Number of Units slot. When first configuring this method, the user must enter the Number of Units, then run the model (stepping through 1 timestep is enough) to grow the slots to the right dimensions.

The following slots are instantiated when this method is selected:

☛ Unit Avoidance Zones Table

Type: TableSlot

Units: Length, Power, Power

Description: This table represents zones in the Unit Power Table that should be avoided for operations.

Information: The avoidance zone table has one block per unit and three columns per block: head, power at bottom of the avoidance zone, power at the top of the avoidance zone. This table effectively removes regions from the Unit Power Table. The regions removed might have to be interpolated between points in the table. Heads that appear in this table must appear in the Unit Power Table as well.

I/O: Required Input

Links: NA

Unit 1			Unit 2		
Operating Head (ft)	Min. Power at Zone Bottom (MW)	Max. Power at Zone Top (MW)	Operating Head (ft)	Min. Power at Zone Bottom (MW)	Max. Power at Zone Top (MW)
100	4.5	6	100	8	9
200	5	6.4	200	9	9.5
300	6	7	300	9.2	9.9

Method Details

In Simulation, the dispatch method will execute this method once head and power are computed for each unit. This method will determine if the computed head and power fall inside an avoidance zone. If so, the method will issue an error but not stop the run. If the method is called and there is no valid Unit Power, then the method is exited without performing any computations.

23.1.38 Frequency Regulation

This category depends on selecting the Unit Power Table method, although in the future it might be enabled for other power methods. The frequency regulation methods model the provision of the frequency regulation ancillary service, that is, how the reservoir can be made available to flexibly follow a load demand within a specified range during a certain period in order to affect the frequency of the generated power.

23.1.38.1 No Method (default)

This is the default, do-nothing method; no regulation is modeled.

23.1.38.2 Unit Frequency Regulation

NOTE: ALTHOUGH THE METHOD CAN BE SELECTED AND SLOTS ARE ADDED, THIS METHOD IS NOT YET IMPLEMENTED.

When frequency regulation is scheduled, it allows the unit to follow the real time load. Exactly what will happen in real time is unknowable. This results in two sets of values at scheduling time, nominally scheduled power and turbine release. It is uncertain if the real time operators will actually use the service. At present, we distinguish between the nominal "scheduled" power (and turbine release) that the regulation is allowed to depart from and the "expected" power generation (and turbine release) that will take place when regulation is allowed. Both are important. The scheduled power sets the baseline for regulation and should be communicated to the power dispatchers. The expected power and release are more useful for coordinating a plant with the rest of the system.

Slots Added by this Method

Note, many of these slots have column or row dimensions based on the number of units. The rows/columns of these slots are expanded at the beginning of the run to match the value in the Number of Units slot. When first configuring this method, the user must enter the Number of Units, then run the model (stepping through 1 timestep is enough) to grow the slots to the right dimensions.

The following slots are instantiated when this method is selected:

☛ **Unit Regulation Table**

Type: TableSlot
Units: Length, Flow, Power, Flow, Power
Description: This table (not visible to the user) represents the available regulation (both up and down in terms of flow and power) for each unit at each point in the Unit Power Table.
Information: This table is calculated using data in the Unit Power Table and the Avoidance Zone Table (if applicable) and could be calculated automatically at beginning of run in simulation and/or optimization. This table consists of a block of six

columns for each unit. The head and flow values should be the same as the Unit Power Table. The other four columns in the block are respectively Regulation Flow Up, Regulation Power Up, Regulation Flow Down, and Regulation Power Down. These values represent the minimum and maximum power achievable from the initial flow value without passing through an avoidance zone.We require that the heads in this table appear in the Unit Power Table as well.

I/O: Automatically calculated at beginning of run

Links: NA

Unit Two Sided Regulation

Type: AggSeriesSlot
Units: Power
Description: The value is the two sided frequency regulation for the unit at that timestep.
Information: There is one column for each unit.
I/O: Input or Output
Links: NA

Unit Regulation Up

Type: AggSeriesSlot
Units: Power
Description: The value is the frequency regulation up for the unit at that timestep.
Information: There is one column for each unit.
I/O: Input or Output
Links: NA

Unit Regulation Down

Type: AggSeriesSlot
Units: Power
Description: The value is the frequency regulation down for the unit at that timestep.
Information: There is one column for each unit.
I/O: Input or Output
Links: NA

Unit Possible Regulation Up

Type: AggSeriesSlot
Units: Power
Description: The value is the possible regulation up for the unit at that timestep.
Information: There is one column for each unit.
I/O: Input or Output
Links: NA

☛ Unit Possible Regulation Down

Type: AggSeriesSlot
Units: Power
Description: The value is the possible regulation down for the unit at that timestep.
Information: There is one column for each unit.
I/O: Input or Output
Links: NA

☛ Unit Flow Addition For Regulation

Type: AggSeriesSlot
Units: Flow
Description: The value is the additional release required to reach the frequency high point for the unit at that timestep.
Information: There is one column for each unit. This value is typically returned from optimization and set via a rule.
I/O: Rule
Links: NA

☛ Unit Flow Reduction For Regulation

Type: AggSeriesSlot
Units: Flow
Description: The value is the reduction in release required to reach the frequency low point for the unit at that timestep.
Information: There is one column for each unit. This value is typically returned from optimization and set via a rule.
I/O: Rule
Links: NA

☛ Unit Scheduled Mechanical Power

Type: AggSeriesSlot
Units: Power
Description: The value is the scheduled mechanical power generation, before subtracting regulation (or reactive power) for the unit at that timestep.
Information: There is one column for each unit.
I/O: Output only
Links: NA

☛ Unit Scheduled Turbine Release

Type: AggSeriesSlot
Units: Flow
Description: The value is the turbine flow which corresponds to the Unit Scheduled mechanical Power for the unit at that timestep.

Information: There is one column for each unit.
I/O: Output only
Links: NA

☛ Unit Operating Cost Per Regulation Table
Type: TableSlot
Units: VALUE ($)
Description: For each generating unit, this is the cost per unit of regulation.
Information: There is one row for each unit.
I/O: Input
Links: NA

☛ Unit Operating Cost
Type: AggSeriesSlot
Units: VALUE ($)
Description: This is the total cost of using a unit for regulation incurred during the run.
Information: There is one column for each unit.
I/O: Output
Links: NA

☛ Operating Cost
Type: AggSeriesSlot
Units: VALUE ($)
Description: The value is the sum of the unit operating costs.
Information: This is an existing slot with only one column.
I/O: Output only
Links: NA

☛ Regulation
Type: SeriesSlot
Units: POWER
Description: Total regulation for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Two Sided Regulation.
I/O: Output only
Links: Optional

☛ Plant Regulation Up
Type: SeriesSlot
Units: POWER
Description: Total regulation up for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Regulation Up.
I/O: Input or Output

Links: Linkable, typically to the Thermal object System Regulation Up

☛ PLANT REGULATION DOWN

Type: SeriesSlot
Units: POWER
Description: Total regulation down for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Regulation Down.
I/O: Input or Output
Links: Linkable, typically to the Thermal object System Regulation Down

☛ PLANT POSSIBLE REGULATION UP

Type: SeriesSlot
Units: POWER
Description: Total possible regulation up for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Possible Regulation Up.
I/O: Output only
Links: No

☛ PLANT POSSIBLE REGULATION DOWN

Type: SeriesSlot
Units: POWER
Description: Total possible regulation down for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Possible Regulation Down.
I/O: Output only
Links: Optional

☛ PLANT FLOW ADDITION FOR REGULATION

Type: SeriesSlot
Units: FLOW
Description: Total additional turbine release required in order to reach the frequency regulation high point for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Flow Addition For Regulation.
I/O: Output only
Links: Optional

☛ PLANT FLOW REDUCTION FOR REGULATION

Type: SeriesSlot
Units: FLOW
Description: Total reduction in turbine release required in order to reach the frequency regulation low point for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Flow Reduction For Regulation.

I/O: Output only

Links: No

☛ Plant Scheduled Mechanical Power

Type: SeriesSlot
Units: POWER
Description: Total scheduled mechanical power for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Scheduled Mechanical Power.
I/O: Output only

Links: No

☛ Plant Scheduled Turbine Release

Type: SeriesSlot
Units: FLOW
Description: Total scheduled turbine release for the reservoir (plant) at that timestep.
Information: This value is the sum over the columns of Unit Scheduled Turbine Release.
I/O: Output only

Links: No

In Simulation, the Unit Power Table method will execute this method when the Unit Turbine Release is known. At this time, this method cannot be called if Unit Energy is specified (input or rules):

- Unit Scheduled Turbine Release[t,u] = Unit Turbine Release[t,u] - (Unit Flow Addition for Regulation[t,u] + Unit Flow Reduction For Regulation [t,u])/2
- Unit Scheduled Mechanical Power[t,u] = Unit Power Table(head[t], Unit Scheduled Turbine Release[t,u])
- Unit Regulation Up[t.u] = Unit Power Table(head[t], Unit Scheduled Turbine Release[t,u] + Unit Flow Addition for Regulation[t,u] /2)
- Unit Regulation Down [t,u] = Unit Power Table(head[t], Unit Scheduled Turbine Release[t,u]-Unit Flow Reduction For Regulation [t,u]/2)
- Unit Regulation [t,u] = max(Unit Regulation Up[t.u] , Unit Regulation Down [t,u])
- Unit Power[t,u] = Unit Scheduled Mechanical Power[t,u] + Unit Regulation Up[t.u]/2 - Unit Regulation Down [t,u]/2
- Calculate Unit Operating Cost[t,u] = timestep * Unit Operating Cost Per Regulation[u] * (Unit Regulation Up[t,u] + Unit Regulation Down[t,u])
- Compute plant level values as a sum of unit values

23.1.39

23.2 Dispatch Methods

23.2.1 solveMB_givenInflowHW

This dispatch method is placed on the queue when the list of knowns/unknowns (below) is met. It solves for Outflow and Energy.

Required Knowns

- Diversion
- Return Flow
- Inflow2
- Inflow
- Pool Elevation

Required Unknowns

- Outflow
- Energy
- Storage

First, this method checks if the Pool Elevation at the current timestep is a Target flag. If so, it begins the target solving process. Otherwise, it checks if the Canal Flow slot is linked. If it is, the method exits, and waits for the Canal to solve. The storage corresponding to a flat pool elevation is found as a seed value for the calculations via the Elevation Volume table. Since this is a Slope Power Reservoir, this storage is not the actual storage in the reservoir.

If these checks pass, the calculations which depend on the Pool Elevation are done (evaporation and seepage), if selected. The slope storage calculations are then carried out, including the mass balance:

$$Outflow = Storage(-1) - Storage + Inflow$$

This equation may or may not contain other elements reflecting the user method choices (including hydrologic inflow, evaporation, precipitation, seepage, and bankstorage). This is an iterative process between the mass balance, the wedge storage calculation, and the bank storage calculation, if any. Spill and release are then found according to the user selected methods and the equation:

$$Outflow = Spill + Release$$

The power calculation, energy in storage, spilled energy, and future value calculation user methods are called. The dispatch method is then complete.

23.2.2 solveMB_givenOutflowHW

This dispatch method is placed on the queue when the list of knowns/unknowns (below) is met. It solves for Inflow.

Required Knowns

- Diversion
- Return Flow
- Inflow2
- Outflow
- Pool Elevation

Required Unknowns

- Inflow
- Storage

First, this method finds the Storage associated with the known Pool Elevation. It then looks to see if the Pool Elevation at the current timestep is a Target Op. If so, it begins the target solving process. Otherwise, it checks if the Canal Flow slot is linked. If it is, the method exits, and waits for the Canal to solve.

If these checks pass, the Outflow is checked for the Max Capacity flag. If this flag is present, the maximum outflow is found from the given Pool Elevation. The calculations which depend on the Pool Elevation are done (evaporation and seepage), if selected. The slope storage calculations are then carried out, including the mass balance:

$$Inflow = Storage - Storage(-1) + Outflow$$

This equation may or may not contain other elements reflecting the user method choices (including hydrologic inflow, evaporation, precipitation, seepage, and bankstorage). This is an iterative process between the mass balance, the wedge storage calculation, and the bank storage calculation, if any. Spill and release are then found according to the user selected methods and the equation:

$$Outflow = Spill + Release$$

The power calculation, energy in storage, spilled energy, and future value calculation user methods are called. The dispatch method is then complete.

23.2.3 solveMB_givenInflowStorage

This dispatch method is placed on the queue when the list of knowns/unknowns (below) is met. It solves for Outflow and Energy.

REQUIRED KNOWNS

- **DIVERSION**
- **RETURN FLOW**
- **INFLOW2**
- **INFLOW**
- **STORAGE**

REQUIRED UNKNOWNS

- **OUTFLOW**
- **ENERGY**
- **POOL ELEVATION**

First, this method checks if the Storage at the current timestep is a Target Op. If so, it begins the target solving process. Otherwise, it checks if the Canal Flow slot is linked. If it is, the method exits, and waits for the Canal to solve.

If these checks pass, the mass balance is carried out as follows:

$$Outflow = Storage(-1) - Storage + Inflow$$

This equation may or may not contain other elements reflecting the user method choices (including hydrologic inflow, evaporation, precipitation, seepage, and bankstorage). The slope storage, mass balance, evaporation, and seepage calculations are combined in an iterative process to arrive at a final value for the Pool Elevation and Outflow. Spill and release are then found according to the user selected methods and the equation:

$$Outflow = Spill + Release$$

The power calculation, energy in storage, spilled energy, and future value calculation user methods are called. The dispatch method is then complete.

23.2.4 solveMB_givenOutflowStorage

This dispatch method is placed on the queue when the list of knowns/unknowns (below) is met. It solves for Inflow.

Required Knowns

- ☛ **Diversion**
- ☛ **Return Flow**
- ☛ **Inflow2**
- ☛ **Outflow**
- ☛ **Storage**

Required Unknowns

- ☛ **Inflow**
- ☛ **Pool Elevation**

First, this method finds the Pool Elevation associated with the known Storage. It then looks to see if the Storage at the current timestep is a Target Op. If so, it begins the target solving process unless the Outflow is flagged Max Capacity. This case will result in an error. Otherwise, it checks if the Canal Flow slot is linked. If it is, the method exits, and waits for the Canal to solve.

If these checks pass, the Outflow is checked for the Max Capacity flag. If this flag is present, the maximum outflow is found from the given Pool Elevation, then the mass balance is carried out:

$$Inflow = Storage - Storage(-1) + Outflow$$

This equation may or may not contain other elements reflecting the user method choices (including hydrologic inflow, evaporation, precipitation, seepage, and bankstorage). Spill and release are then found according to the user selected methods and the equation:

$$Outflow = Spill + Release$$

The energy in storage, spilled energy, and future value calculation user methods are called. The dispatch method is then complete.

23.2.5 solveMB_givenInflowOutflow

This dispatch method is placed on the queue when the list of knowns/unknowns (below) is met. It solves for Storage (and Pool Elevation).

Required Knowns

- **Diversion**
- **Return Flow**
- **Inflow2**
- **Inflow**
- **Outflow**

Required Unknowns

- **Pool Elevation**
- **Storage**

First, this method checks if there are Pool Elevation and Storage values for the previous timestep. If these slots are valid the method continues and checks if a Canal Object is linked. If there is a Canal Object and it has not yet solved, the method exits and waits for the Canal Object to solve.

If these checks pass, Outflow is checked for the Max Capacity flag. If this flag is present, the maximum outflow is found from the mass balance and spill methods using a convergence algorithm. The convergence algorithm used is detailed HERE (Appendix A: Reservoir Convergence). If this flag is not present, then the mass balance is carried out as follows:

$$Storage = Storage(-1) + Inflow - Outflow$$

This equation may or may not contain other elements reflecting the user method choices (including hydrologic inflow, evaporation, precipitation, seepage, and bankstorage). This solution is performed with an iterative process. (See note below.) Spill and Release are then found according to the user selected methods and the equation:

$$Outflow = Spill + Release$$

The energy in storage, spilled energy, and future value calculation user methods are called. The dispatch method is then complete.

NOTE: In specific cases when Storage is very close to zero, the iterative mass balance solution will follow one of two possible processes. The first process allows the loop to iterate in the negative storage range before final convergence. Convergence in this situation is typically quite rapid, and in instances when the final storage is, in fact, negative, an error is posted stating that the outflow is too large to be physically possible. This process is invoked if an additional row is appended to the Elevation Volume table specifying a negative storage value within which the loop can iterate. If specifying and allowing negative storage values in the iteration is not desired, no negative storage value should be appended to the Elevation Volume table. In this second process, the algorithm uses storage equals zero whenever it is in the negative storage range. If the outflow is really too great to be physically possible, the algorithm will keep iterating until it reaches maximum iterations. If this happens, RiverWare does a final mass balance check at the storage equals zero point and posts an error stating that the outflow is too large.

23.2.6 solveMB_givenEnergyInflow

This dispatch method is placed on the queue when the list of knowns/unknowns (below) is met. It solves for Outflow.

REQUIRED KNOWNS

☛ Diversion

☛ Return Flow

☛ Inflow2

☛ Inflow

☛ Energy

Required Unknowns

☛ Pool Elevation

☛ Storage

An error will result if this object is linked to a Canal and this dispatch method is executed. A pre-iteration spill calculation and power release calculation are done next. A double iteration is then done with the inner loop iterating evaporation, bankstorage, seepage and mass balance (including slope storage) to solve for Storage, and the outer loop iterating spill and power (tailwater, and power release) to solve for Outflow. (See note below.)

$$Storage = Storage(-1) + Inflow - Outflow$$

$$Outflow = Spill + Release$$

The power, energy in storage, spilled energy, and future value calculation user methods are called. The dispatch method is then complete.

NOTE: In specific cases when Storage is very close to zero, the iterative mass balance solve Storage solution will follow one of two possible processes. The first process allows the loop to iterate in the negative storage range before final convergence. Convergence in this situation is typically quite rapid, and in instances when the final storage is, in fact, negative, an error is posted stating that the outflow is too large to be physically possible. This process is invoked if an additional row is appended to the Elevation Volume table specifying a negative storage value within which the loop can iterate. If specifying and allowing negative storage values in the iteration is not desired, no negative storage value should be appended to the Elevation Volume table. In this second process, the algorithm uses storage equals zero whenever it is in the negative storage range. If the outflow is really too great to be physically possible, the algorithm will keep iterating until it reaches maximum iterations. If this happens, RiverWare

does a final mass balance check at the storage equals zero point and posts an error stating that the outflow is too large.

23.2.7 solveMB_givenEnergyStorage

This dispatch method is placed on the queue when the list of knowns/unknowns (below) is met. It solves for Inflow.

Required Knowns

- ☛ **Diversion**
- ☛ **Return Flow**
- ☛ **Inflow2**
- ☛ **Storage**
- ☛ **Energy**

Required Unknowns

- ☛ **Pool Elevation**
- ☛ **Inflow**

Dispatching with this method will result in an error if the Storage is a target or if the Canal Flow slot is linked. Pre-iteration spill and power (tailwater and power release) calculations are done. A double iteration is performed iterating slope storage, seepage and evaporation in the inner loop to solve for Pool Elevation, and mass balance (solving for Inflow), spill, and power (tailwater and power release) in the outer loop to solve for Outflow.

$$Outflow = Spill + Release$$

$$Inflow = Storage - Storage(-1) + Outflow$$

The power, energy in storage, spilled energy, and future value calculation user methods are called. The dispatch method is then complete.

23.2.8 solveMB_givenEnergyHW

This dispatch method is placed on the queue when the list of knowns/unknowns (below) is met. It solves for Inflow.

Required Knowns

- Diversion
- Return Flow
- Inflow2
- Pool Elevation
- Energy

Required Unknowns

- Inflow
- Storage

Dispatching with this method will result in an error if the Pool Elevation is a target or if the Canal Flow slot is linked. This should be avoided by insuring that the knowns/unknowns will not match this dispatch method during the target period or when a Canal is linked. Pre-iteration spill, power (tailwater and power release), evaporation and seepage calculations are calculated. A double iteration is performed, iterating slope storage, seepage and evaporation in the inner loop to solve for Pool Elevation, and mass balance (solving for Inflow), spill, and power (tailwater and power release) in the outer loop to solve for Outflow.

$$Outflow = Spill + Release$$

$$Inflow = Storage - Storage(-1) + Outflow$$

The power, energy in storage, spilled energy, and future value calculation user methods are called. The dispatch method is then complete.

23.2.9 solveMB_givenInflowRelease

This dispatch method is placed on the queue when the list of knowns/unknowns (below) is met. It solves for Storage, Pool Elevation, Energy, and Outflow.

Required Knowns

☛ Diversion

☛ Return Flow

☛ Inflow

☛ Inflow2

☛ Turbine Release

Required Unknowns

☛ Energy

☛ Pool Elevation

☛ Storage

☛ Outflow

This method exits with an error if Canal Flow slot is linked or if Spill is input.

If the Turbine Release is flagged Unit Values (U), the Unit Turbine Release subslots are summed to calculate and set the Turbine Release slot. If all the Unit Turbine Release slots are NaN, an error is issued. If the Unit Turbine Release slot is not visible because the Unit Power Table method is not selected, an error is issued.

If there is an unregulated spillway crest (because an unregulated spill method is selected) the method computes the upper limit for unregulated spill to prevent spilling too much water to drop the reservoir below the crest. Typically, this limit is computed as the volume of water above the crest converted to a flow. See the Unregulated Spill method for details.

The Outflow is computed as the sum of turbine release and spill inputs (if any) according to the user selected methods and the equation:

$$Outflow = Spill + Release$$

Then the Storage (including wedge storage) is found with the equation:

$$Storage = Storage(-1) + Inflow - Outflow$$

This equation may or may not contain other elements reflecting the user method choices (including hydrologic inflow, evaporation, precipitation, seepage, and bankstorage). The solution is done with an iterative process. (See note below.) Any error in the mass balance added to the unregulated spill slot, if it is being used. This value for unregulated spill may not conform to the unregulated spill table. Care should be exercised when dispatching using this method to insure that the mass balance is maintained.

The power, energy in storage, spilled energy, and future value calculation user methods are called. The dispatch method is then complete.

NOTE: In specific cases when Storage is very close to zero, the iterative mass balance solution will follow one of two possible processes. The first process allows the loop to iterate in the negative storage range before final convergence. Convergence in this situation is typically quite rapid, and in instances when the final storage is, in fact, negative, an error is posted stating that the outflow is too large to be physically possible. This process is invoked if an additional row is appended to the Elevation Volume table specifying a negative storage value within which the loop can iterate. If specifying and allowing negative storage values in the iteration is not desired, no negative storage value should be appended to the Elevation Volume table. In this second process, the algorithm uses storage equals zero whenever it is in the negative storage range. If the outflow is really too great to be physically possible, the algorithm will keep iterating until it reaches maximum iterations. If this happens, RiverWare does a final mass balance check at the storage equals zero point and posts an error stating that the outflow is too large.

24. Storage Reservoir

The Storage Reservoir is the least complicated of all the reservoirs modeled in **RiverWare**™. The only process performed by the Storage Reservoir is the storage of water. No power production facilities exist on the Storage Reservoir. The following slots are those that are general to the object. This means that these slots will exist on the Storage Reservoir regardless of the selected user methods.

General Slots

(slots which always appear for this object)

☞ Canal Flow

Type: Series Slot
Units: FLOW
Description: flow into (out of) the reservoir from (to) a canal
Information:
I/O: Output only
Links: May be linked to either the Flow 1 or Flow 2 slot of the Canal object. If not linked, the slot is set to zero.

☞ Convergence Percentage

Type: Table
Units: NONE
Description: A percentage value ranging from 0 to 1 used for convergence in all iterative calculations
Information: Click **HERE (Appendix A: Reservoir Convergence)** for more information on the convergence algorithm
I/O: Optional; defaults to 0.0001 if not input.
Links: Not linkable

☞ Diversion

Type: Series
Units: FLOW
Description: flow from the reservoir to a diverting object
Information: If not linked or input it is set to zero.
I/O: Optional; may be input or linked or neither

Links: May be linked to the Total Diversion slot on an Agg Diversion Site or the Total delivery Request slot on an AggDistribution Canal

☞ Diversion Capacity

Type: Scalar Slot
Units: FLOW
Description: used to hold the maximum diversion physically possible from the reservoir
Information: This slot is used in the accounting system for allocation purposes and can be used in Rulebased Simulation
I/O: Input only
Links: Not linkable

☞ Elevation Volume Table

Type: Table
Units: LENGTH vs. VOLUME
Description: reservoir pool elevation vs. reservoir storage
Information:
I/O: Required input
Links: Not linkable

☞ Flow FROM Pumped Storage

Type: Series Slot
Units: FLOW
Description: flow into the reservoir from a pumped storage reservoir
Information:
I/O: Optional; usually linked if used.
Links: May be linked to the Outflow slot of a Pumped Storage object.

☞ Flow TO Pumped Storage

Type: Series Slot
Units: FLOW
Description: flow out of the reservoir into a pumped storage reservoir
Information:
I/O: Optional; usually linked if used.
Links: May be linked to the Pumped Flow slot of a Pumped Storage object.

☞ Inflow

Type: Multi Slot
Units: FLOW
Description: inflow into the reservoir from upstream

Information:
I/O: Optional; if not input by the user, it is set through either mass balance computations or the propagation of values across the link.
Links: May be linked to one or more outflow slots of upstream objects.

☞ Max Iterations

Type: Table
Units: NOUNITS
Description: maximum number of allowable iterations for iterative loops in the solution algorithms
Information: Used in conjunction with Convergence Percentage as a stopping criterion for iterative calculations.
I/O: Optional; defaults to 100 if not input.
Links: Not linkable

☞ Max Release

Type: Table
Units: LENGTH vs. FLOW
Description: reservoir pool elevation vs. maximum release through controlled outlet-conduit of a reservoir (excluding spill)
Information:
I/O: Required input
Links: Not linkable

☞ Outflow

Type: Series Slot
Units: FLOW
Description: outflow from reservoir
Information: The Outflow from a Storage Reservoir is equal to the sum of the Release and the Spill.
I/O: Optional; if not input by the user, it is set through either the mass balance computations or the propagation of values across the link.
Links: May be linked to the inflow slot of a downstream object.

☞ Pool Elevation

Type: Series Slot
Units: LENGTH
Description: elevation of the water surface in the reservoir
Information: There must be an initial value for either Storage or Pool Elevation given by the user for the first timestep.

I/O: Optional; if not input by the user, it is solved by the mass balance computations. It may take a TARGET flag indicated by the user for target operation solution.

Links: May be linked to Tailwater Elevation or Tailwater Base Value of an upstream object or to Elevation 1 or Elevation 2 of a Canal object.

☞ Release

Type: Series Slot
Units: FLOW
Description: flow through the outlet-conduit of a storage reservoir (excluding spill)
Information:
I/O: Optional; solved for if not input.
Links: Usually not linked

☞ Return Flow

Type: Multi Slot
Units: FLOW
Description: flow returning from a diverting object
Information:
I/O: Optional; defaults to zero if not linked or input.
Links: May be linked to one or more Return Flow slots on Water User objects or the Total Return Flow slot on the Agg Diversion Site objects.

☞ Spill

Type: Series Slot
Units: FLOW
Description: sum of the regulated and unregulated spills and bypass
Information:
I/O: Optional; may be input or solved for by **RiverWare**™ (see spill calculation methods).
Links: Usually not linked

☞ Storage

Type: Series Slot
Units: VOLUME
Description: volume of water stored in the reservoir
Information: This slot may be flagged as a TARGET storage value by the user. If flagged as a TARGET, a target operation solution is used.
I/O: Optional; if not input by the user, it is set through mass balance computations.
Links: Usually not linked

☞ TOTAL INFLOWS

Type: Series
Units: FLOW
Description: Summary slot displaying the flows into and out of the reservoir excluding the flows through the outlet works
Information: Total Inflows is calculated using the following equation:

$$\text{Total Inflows} = \text{Inflow} + \text{Canal Flow} + \text{Hydrologic Inflow} + \text{Hydrologic Inflow Adjust} + \text{Hydrologic Inflow Forecast} + \text{Return Flow} + \text{Flow FROM Pumped Storage} - \text{Flow TO Pumped Storage} - \text{Diversion}$$

Any component that is not in use or is not valid defaults to zero.

I/O: Output only
Links: Not linkable

☞ INFLOW SUM

Type: Series
Units: FLOW
Description: Sum of the total flows entering the reservoir at each timestep
Information: Inflow Sum is calculated using the following equation:

$$\text{Inflow Sum} = \text{Inflow} + \text{Canal Flow} + \text{Hydrologic Inflow} + \text{Hydrologic Inflow Adjust} + \text{Hydrologic Inflow Forecast} + \text{Return Flow} + \text{Flow FROM Pumped Storage}$$

I/O: Output only
Links: Not Linkable

Note: Either the initial (Beginning of Run) value for Pool Elevation or Storage must be input by the user.

24.1 User Methods

24.1.1 Energy in Storage

The methods available in the Energy in Storage category are used to calculate the total energy that could be produced by the water stored in the reservoir.

24.1.1.1 None

Chosen if the user does not want to calculate the Energy In Storage. No slots are specifically associated with this method. This method performs no calculations.

24.1.1.2 EIS Table Lookup

The EIS TableLookup method obtains the amount of Energy In Storage from a table of Pool Elevation vs. Energy In Storage values and the Pool Elevation.

Slots Specific to this Method

☞ **Energy In Storage**

Type: AggSeriesSlot
Units: ENERGY
Description: Energy In Storage in the Reservoir
Information:
I/O: Output only
Links: Usually not linked

☞ **Energy In Storage Table**

Type: TableSlot
Units: LENGTH vs. ENERGY
Description: Pool Elevation vs. Energy In Storage In the Reservoir
Information:
I/O: Required input
Links: Not linkable

The calculations involved with this method are very simple. The Pool Elevation for the current timestep is used to determine the Energy In Storage using the Energy In Storage Table. Simple linear interpolation is used.

24.1.1.3 EIS Table Lookup with Cons Pool

This method will be available only when the **Conservation Pool** or **Conservation and Flood Pools** method in the **Operating Levels** category is selected.

SLOTS SPECIFIC TO THIS METHOD

☞ **ENERGY IN STORAGE**

Type: AggSeriesSlot
Units: ENERGY
Description: Energy In Storage in the Reservoir
Information:
I/O: Output only
Links: Usually not linked

☞ **ENERGY IN STORAGE TABLE**

Type: TableSlot
Units: LENGTH VS. ENERGY
Description: Pool Elevation vs. Energy In Storage In the Reservoir
Information:
I/O: Required input
Links: Not linkable

☞ **CONSERVATION POOL FULL EIS**

Type: Series Slot
Units: MWH
Description: The EIS at the Top of the Conservation Pool
Information:
I/O: Output Only
Links: Not Linkable

The method is executed at the end of each dispatch method.

The **Pool Elevation** for the timestep is looked up in **Energy In Storage Table**. Simple linear interpolation is used. The resulting **Energy in Storage** is then set on the slot.

Next, the **Conservation Pool Full EIS** is calculated as follows:

$$\text{Top Conservation Pool Elevation}[t] = \text{Operating Level Table}[t, \text{Top of Conservation Pool Level}]$$

$$\text{Conservation Pool Full EIS}[t] = \text{Energy in Storage Table}[\text{Top of Conservation Pool Elevation}[t]]$$

Note, in the equation, the appropriate Operating Level Table will be used based on the timestep and the computation in the selected method in the **Conditional Operating Levels** category.

24.1.2 Tailwater

The Tailwater category is used to model the effects of the tailwater elevation on reservoir releases. The tailwater elevation represents the water surface elevation immediately downstream of the storage reservoir's release structure.

24.1.2.1 None

This is the default method. It performs no calculations and there are no slots associated with it.

24.1.2.2 Input Tailwater Elevation

This method is used if the tailwater elevation is linked to the downstream reservoir pool elevation or directly input by the user. There are no tailwater elevation calculations associated with this method. In other words, the outflow does not affect the tailwater elevation other than by changing the downstream pool elevation. When the tailwater elevation is above the tailwater reference elevation, the effective head is used to compute the max release.

Slots Specific to this Method

☞ Tailwater Elevation

Type: Series Slot
Units: LENGTH
Description: The elevation associated with the downstream end of the head drop across a controlled release structure.
Information: The Tailwater Elevation is usually considered to be the pool elevation of the downstream reservoir.
I/O: Optional; may be linked or input
Links: Usually linked to the Pool Elevation of the downstream reservoir

☞ Tailwater Reference Elevation

Type: Scalar Slot
Units: LENGTH
Description: The lowest reservoir discharge elevation where there are no effects from a downstream reservoir pool elevation
Information: This slot is the elevation of the downstream side of the controlled release structure. The Effective Head is computed using either this elevation or the Tailwater Elevation (whichever is greater).
I/O: Required Input
Links: Not linkable

☞ Effective Head

Type: Series Slot
Units: LENGTH
Description: The head difference across a controlled release structure
Information: Computed as difference between the reservoir pool elevation the greater value of either the Tailwater Elevation or the Tailwater Reference Elevation. If the Tailwater Reference Elevation is greater than the Tailwater Elevation, the Effective Head slot is not used.
I/O: Output only
Links: Not linkable

☞ Head Vs Max Release

Type: Table Slot
Units: LENGTH vs FLOW
Description: The relationship between effective head and maximum reservoir release
Information: When the Tailwater Elevation is greater than the Tailwater Reference Elevation, the Effective Head is used with this table to determine the max release. Otherwise, the Pool Elevation and Max Release table are used to compute the Max Release (i.e. the Effective Head is not considered).
I/O: Required Input
Links: Not linkable

In this method, Tailwater Elevation is either linked to the downstream reservoir pool elevation, or input by the user. Therefore, there are no calculations to determine the Tailwater Elevation. However, the Tailwater Elevation is used to compute the Effective Head, which is used to compute the maximum reservoir release. If the Tailwater Elevation is greater than the Tailwater Reference Elevation, the Effective Head is computed as:

$$\text{Effective Head} = \text{Pool Elevation} - \text{Tailwater Elevation}$$

If the Tailwater Elevation is less than the Tailwater Reference Elevation, Effective Head is computed as:

$$\text{Effective Head} = \text{Pool Elevation} - \text{Tailwater Reference Elevation}$$

Note: Important Max Release Calculation Details:

If a tailwater method is not selected, the maximum reservoir release is computed using the average Pool Elevation and the Max Release table. The selection of a tailwater method allows the modification of the maximum reservoir release calculation. When the Input Tailwater Elevation method is selected, the average Effective Head (using the beginning of timestep and end of timestep values) is used with the Head Vs Max Release table to determine the maximum reservoir release. However, this is done ONLY if BOTH the beginning of timestep and end of timestep Tailwater Elevations are greater than the Tailwater Reference Elevation. Otherwise, the average Pool Elevation is used with the Max Release table to determine the maximum reservoir release.

24.1.3 Spill

The Spill methods (except the Monthly Spill which is described in Section 24.1.3.2) calculate the Spill from Reservoirs based on several possible physical combinations of controlled and uncontrolled spillways.

The **Regulated Spill** and **Bypass** slots are regulated (i.e. controlled) spill structures. Values in these two slots can be specified by the user via inputs or rules. Each slot accommodates spill up to the maximum amount as specified by its rating table (**Regulated Spill Table** and **Bypass Table**). **Unregulated Spill** is an uncontrolled spill. Therefore, it is always a computed output based on the average **Pool Elevation** of the reservoir as specified in the **Unregulated Spill Table**. Thus, the user selects a **Spill** method based on the combination of structures (Regulated Spill, Bypass, and/or Unregulated Spill) that exist on the reservoir and the level of granularity desired.

The total **Spill** slot is the sum of the individual spills from each structure. Spills are calculated twice for each timestep. The first time a **Spill** method is called from a dispatch method, it checks for user inputs, calculates any **Unregulated Spill**, and sets the spill to zero for **Regulated Spill** and **Bypass** structures where there is no user-specified value. The total **Spill** is then calculated and returned to the dispatch method. The dispatch method determines (**Turbine**) **Release** by subtracting the **Spill** from **Outflow**, and executes the user-specified power calculation method. (On the power reservoirs, the slot is called **Turbine Release**, on the storage reservoir, the slot is called **Release**. In this description, we use the term (**Turbine**) **Release**.) If the (**Turbine**) **Release** cannot be met in the power calculation method, a second call is made to the spill calculation method. The excess flow is then distributed among the **Regulated Spill** and/or **Bypass** structures which have available capacity. If both **Regulated Spill** and **Bypass** are available, excess spill is typically first discharged through the **Regulated Spill** (except when the "**Bypass, Regulated and Unregulated**" method is selected; for this method the **Bypass** gets spill first).

The optional DRIFT flag is available on the **Regulated Spill** and **Bypass** slots. When the DRIFT flag is set for several sequential timesteps, the method models varying flow through a set spillway gate in response to fluctuations in **Pool Elevation**. The first timestep prior to initializing drift is used to determine a gate index called **Regulated** (or **Bypass**) **Drift Index**. This index is interpolated from the 3-dimensional **Regulated** (or **Bypass**) **Spill Index Table**, which relates **Pool Elevation** to **Spill** for several gate indices. In the subsequent timesteps where the DRIFT flag is set, the same index is used to find the spill value at the current average **Pool Elevation**. The gate index is maintained throughout the selected time period. At each timestep, a new value of spill is calculated for the structure based on the current **Pool Elevation**. Specifying DRIFT is considered an input, and may affect over determination of spill parameters.

24.1.3.1 None

None is used if **Spill** should not be modeled. In this method, the **Spill** slot on the reservoir is set equal to zero. All releases must be through the (**Turbine**) **Release**.

SLOTS SPECIFIC TO THIS METHOD

☞ NONE

This method sets the **Spill** to zero and performs no further calculations. If the method is being called for the second time in a timestep because there is excess outflow that won't fit through (**Turbine**) **Release**, an error will be posted which states, "No Spillways Available." In this case, either decrease the **Outflow** or select a different spill method.

24.1.3.2 Monthly Spill

The Monthly Spill method is only appropriate for use in long timestep models where Reservoir fluctuations over the timestep cannot be accurately determined. It is important to note that there is no physical (head dependent) basis to the spill in this method. In this method, there are three components to the spill: unregulated, regulated and bypass. Both Regulated Spill and Bypass are considered controlled releases. A Maximum Controlled Release must be specified by the user.

$$Maximum\ Controlled\ Release = max\ Turbine\ Release + max\ Regulated\ Spill + Bypass$$

Any additional Outflow is immediately categorized as Unregulated Spill.

$$Unregulated\ Spill = Max(0.0, Outflow - Maximum\ Controlled\ Release)$$

Bypass may be specified as a user input. If not input, **RiverWare**™ sets this slot to zero. Regulated Spill is always computed by **RiverWare**™. It is set to zero unless the Reservoir cannot release the Outflow through the Release and Bypass. When this occurs, the additional portion of the Outflow is released through the Regulated Spill.

SLOTS SPECIFIC TO THIS METHOD

☞ BYPASS

Type: Series Slot
Units: FLOW
Description: flow through the Bypass spillway
Information:
I/O: Optional; may be input by the user or set to zero by **RiverWare**™.
Links: Usually not linked

☞ Maximum Controlled Release

Type: Table Slot
Units: FLOW
Description: the maximum amount of Turbine Flow, Regulated Spill, and Bypass
Information: 1X1 table slot
I/O: Required input
Links: Not linkable

☞ Regulated Spill

Type: Series Slot
Units: FLOW
Description: excess Outflow not released through the turbine(s)
Information:
I/O: Output only
Links: Usually not linked

☞ Unregulated Spill

Type: Series Slot
Units: FLOW
Description: Outflow in excess of the Maximum Controlled Release
Information:
I/O: Output only
Links: Not linkable

Initially, the Monthly Spill method is called before Release has been calculated. If the Outflow is greater than the Maximum Controlled Release, Unregulated Spill is set as:

$$Unregulated\ Spill = Outflow - Maximum\ Controlled\ Release$$

If the Outflow is less than the Maximum Controlled Release, Unregulated Spill is set to zero. In both cases, all of the following evaluations are also made:

$$Regulated\ Spill = 0.0$$

$$Bypass = user\ input\ or\ 0.0$$

$$Spill = Unregulated\ Spill + Regulated\ Spill + Bypass$$

After Release is calculated, the Monthly Spill method may be called a second time. The method is called a second time if the Release cannot accommodate the remaining portion of the Outflow:

$$Outflow > Unregulated\ Spill + Release + Bypass$$

Remember that Unregulated Spill was calculated before the Release was calculated.

If this occurs, Regulated Spill and Spill are reevaluated as follows:

$$Regulated\ Spill = Outflow - Unregulated\ Spill - Release - Bypass$$

$$Spill = Unregulated\ Spill + Regulated\ Spill + Bypass$$

24.1.3.3 Unregulated

The Unregulated spill method models a single uncontrolled spillway called **Unregulated Spill**. The **Unregulated Spill** is a function of the average reservoir **Pool Elevation**. Because it is uncontrolled, it takes precedence (i.e. water goes through it first) over other types of outflow (i.e. **Release** or **Turbine Release**) in the reservoir. When this method is chosen the user category Unregulated Spill Type , Section 24.1.4, will appear.

The user may not specify (input or via rules) any spill slots with this method.

Slots Specific to this Method

☞ Unregulated Spill

Type: Series Slot
Units: FLOW
Description: spill corresponding to the average Pool Elevation over the timestep
Information:
I/O: Output only
Links: Not linkable

☞ Unregulated Spill Capacity Fraction

Type: Series Slot
Units: Decimal
Description: The fraction of the Unregulated Spill structure that is available.
Information: If not input or set by a rule, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 50 ft of a 1000 ft long crest is blocked, the Unregulated Spill Capacity Fraction would be input to 0.95.
I/O: Input, set by a rule, or output
Links: Not linkable

☞ Unregulated Spill Table

Type: Table Slot
Units: LENGTH vs. FLOW

Description: Pool Elevation versus corresponding Unregulated Spill values
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required input
Links: Not linkable

When the Unregulated spill method is called for the first time from the Dispatch Method, Unregulated Spill is calculated and Spill is set equal to Unregulated Spill.

THE STEPS FOR COMPUTING UNREGULATED SPILL ARE GIVEN BELOW:

1. A temporary variable called “initHW” is created to represent the Pool Elevation at the beginning of the timestep. Likewise, “endHW” is created to represent the Pool Elevation at the end of the timestep (If the Pool Elevation at the end of the timestep is not known, endHW is set equal to the Pool Elevation at the beginning of the timestep.)

2. The Unregulated Spillway Crest is set equal to the Pool Elevation that corresponds to a Spill of zero (from the Unregulated Spill Table).

3. If both initHW and endHW are less than or equal to the Unregulated Spillway Crest, Unregulated Spill is set equal to zero.

4. If both initHW and endHW are greater than or equal the Unregulated Spillway Crest, the average Pool Elevation is used to determine the Unregulated Spill from the Unregulated Spill Table.

$$\text{Unregulated Spill} = \text{Value from table} \times \text{Unregulated Spill Capacity Fraction}$$

5. If either initHW or endHW is greater than the Unregulated Spillway Crest and the other is lower than the crest, the following evaluations and computations are performed:

$$\text{maxHW} = \text{the greater of initHW and endHW}$$

$$\text{minHW} = \text{the lesser of initHW and endHW}$$

$$\text{avgHW} = \frac{\text{maxHW} + \text{Unregulated Spillway Crest}}{2}$$

$$\text{spill fraction} = \frac{\text{maxHW} - \text{Unregulated Spillway Crest}}{\text{maxHW} - \text{minHW}}$$

That is, spill fraction corresponds to the fraction of the timestep during which spill occurs.

A temporary variable called “temp spill” is obtained from the linear interpolation of the Unregulated Spill Table using avgHW. Unregulated Spill is then calculated as:

$$\text{Unregulated Spill} = \text{spill fraction} \times \text{temp spill} \times \text{Unregulated Spill Capacity Fraction}$$

The Unregulated Spill is then limited to be less than or equal to the maxUnregulatedSpill if one has been calculated. See below.

The Unregulated spill method will be called a second time (after the (**Turbine**) **Release** has been calculated) only if the sum of (**Turbine**) **Release** and **Spill** are less than **Outflow**. When this is the case, an error which reads, “Outflow greater than spillway capacities and Release” is posted because the excess **Spill** cannot be incorporated.

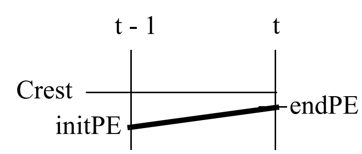
Computing the Maximum Possible Unregulated Spill

If an unregulated spill method is called from the solveMB_givenInflowRelease dispatch method, getMinSpillGivenInflowRelease or solveTurbineRelGivenEnergyInflow RPL predefined functions, there is an upper limit on the unregulated spill. The upper limit prevents the unregulated spill from dropping the reservoir below the crest. The following algorithm is performed at the start of the following methods to compute this maximum unregulated spill (maxUnregulatedSpill):

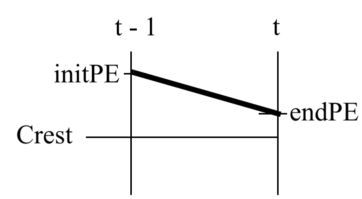
- solveMB_givenInflowRelease dispatch method (Storage and Level Power reservoirs)
- solveMB_givenInflowOutflow dispatch method (Storage and Level Power reservoirs)
- getMinSpillGivenInflowRelease RPL predefined function
- solveTurbineRelGivenEnergyInflow RPL predefined function

1. Given previous Storage and Pool Elevation (initPE), all known inflows (Inflow, Hydrologic Inflow, Precipitation, Return Flow, etc) and known outflows (Release, Regulated Spill, Bypass, Evaporation, Diversion, etc) are used to compute the storage and pool elevation (endPE) that would occur with no additional unregulated spill.

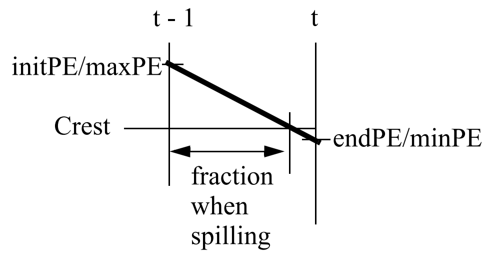
2. If both initPE and endPE are less than or equal to the Unregulated Spillway Crest, maxUnregulatedSpill is set to zero and the computation exits. There is no way there could be unregulated spill. See figure to the right.



3. If both initPE and endPE are greater than or equal to the Unregulated Spillway Crest, the maxUnregulatedSpill is computed as the flow that would draw the reservoir down to exactly reach the crest at the end of the timestep. This computation solves the reservoir mass balance and includes all source and sink terms. All water above the crest could be spilled. See figure to the right.



4. If either initPE or endPE is greater than the Unregulated Spillway Crest and the other is lower than the crest (because of existing Diversions, Evap, etc), the following evaluations and computations are performed: (See figure to the right)



$$maxPE = \text{the greater of initPE and endPE}$$

$$minPE = \text{the lesser of initPE and endPE}$$

$$\text{spill fraction} = \frac{maxPE - \text{Unregulated Spillway Crest}}{maxPE - minPE}$$

That is, spill fraction corresponds to the fraction of the timestep during which spill occurs.

5. The storage at the crest, crestStorage, is computed from the Elevation Volume Table.

6. The storage, maxStorage, that corresponds to the maxPE is found on the Elevation Volume Table.

7. The maxUnregulatedSpill (limited to be greater than or equal to zero) is then computed as:

$$maxUnregulatedSpill = \frac{maxStorage - crestStorage}{TimestepLength} \times spillFraction$$

The maxUnregulatedSpill is applied as a final limit on the Unregulated Spill. Remember, the Unregulated Spill is computed as described in the start of this method using the Unregulated Spill Table and a similar spill fraction approach; it may already be less than the maxUnregulatedSpill.

24.1.3.4 Regulated

The Regulated spill method models **Spill** using one controlled spillway called **Regulated Spill**. Because the spill is controlled, the spill may be any value between zero and the maximum possible regulated spill for that pool elevation. The user may specify (input or via rules) either:

- No slots
- **Spill** or
- **Regulated Spill**

If either is specified and there is excess flow which cannot be met by the (**Turbine**) **Release**, a **RiverWare**™ error will be flagged and the simulation halted.

SLOTS SPECIFIC TO THIS METHOD

☞ Regulated Spill

Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare™**. If Regulated Spill is input by the user and the value is less than the required spill, a **RiverWare™** error is flagged and the simulation is halted.
Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☞ Regulated Spill Capacity Fraction

Type: Series Slot
Units: Decimal
Description: The fraction of the Regulated Spill structure that is available.
Information: If not input or set by a rule, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.
I/O: Input, set by a rule, or output
Links: Not linkable

☞ Regulated Spill Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index
Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Regulated Spill Index Table.
Links: Not linkable

☞ Regulated Spill Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding Maximum Regulated Spill values
Information:
I/O: Required input
Links: Not linkable

☞ Regulated Spill Index Table

Type: Table Slot

Units: NOUNITS vs. LENGTH vs. FLOW

Description: Gate Index vs. Pool Elevation vs. Regulated Spill

Information: Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column 3. The table shown below is an example of the proper way to formulate the Regulated Spill Index Table.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130
4	550	180
4	600	230

I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.

Links: Not linkable

Note: Regulated Spill and Spill are both outputs if neither is specified by the user. Only one of these slots, however, can be specified on a given timestep. When this is the case, the other slot will be output.

The first step in the Regulated spill method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 24.1.5.2. If not, the minRegSpill is zero. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

Release is then checked to see if it has been calculated. If Release is not known, it means that the method is being called for the first time for the particular timestep and the following steps are taken:

1. If both the Spill and Regulated Spill are input (remember setting a DRIFT flag is considered an input), a **RiverWare**™ error is flagged and the run is aborted.

2. If Spill is input by the user, and it is greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted. Otherwise, Regulated Spill is set equal to Spill.

3. If Regulated Spill is input/rules by the user and the DRIFT flag is set, a function is called to perform the drift calculations. A description of the DRIFT function is given at the end of this method description. If Regulated Spill is input and greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted. Otherwise, Spill is set equal to the Regulated Spill.

4. If neither Regulated Spill nor Spill are input, they are both set equal to the minRegSpill.

After Release has been calculated, the Regulated spill function may be called a second time if the sum of Release and Spill is less than the Outflow.

The following calculations and evaluations are performed if the function is called for the second time:

1. If either Spill or Regulated Spill are input, a **RiverWare**™ error is posted stating that **RiverWare**™ is unable to allocate the excess flow and the run is aborted.

2. Regulated Spill is set equal to the Outflow minus the Release. If Regulated Spill is greater than the maxRegSpill, a **RiverWare**™ error is flagged informing the user that Outflow is greater than the spillway capacities and Release and the run is aborted.

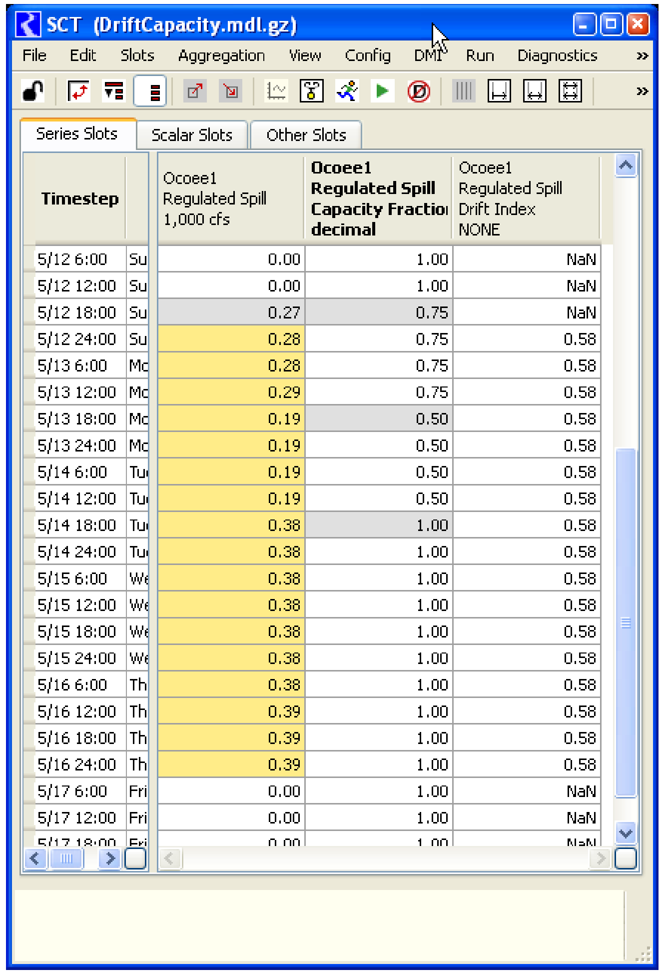
3. Spill is set equal to Regulated Spill.

DRIFT CALCULATIONS:

The drift function is used to calculate Regulated Spill at a specific timestep if it is flagged DRIFT. If the current timestep's Regulated Drift Index is not known, but the previous timestep's Regulated Drift Index is known, the current Regulated Drift Index is set equal to the previous timestep's Regulated Drift Index.

The Drift tables assume that the full spill works is available. Therefore, if there is a Capacity Fraction that is less than 1.0, the Drift calculation (for both regulated and bypass) must be modified.

With the Drift flag is set, if there is a valid Capacity Fraction[t-1] that is not equal to 1.0, then the Capacity Fraction[t] is set to the previous value, but not overwriting inputs or rule values. This causes the Capacity Fraction to remain throughout the drift operation unless it is changed via a new user input. The screenshot to the right shows a sample run. The Capacity fraction is set to 0.75 on 5/12 18:00 and that value remains until a new value is set via user input on 5/13 18:00. Although this set of inputs may not make physical or operations sense, is shows how the algorithm would perform given the inputs shown.



Timestep		Ocoee1 Regulated Spill 1,000 cfs	Ocoee1 Regulated Spill Capacity Fractio decimal	Ocoee1 Regulated Spill Drift Index NONE
5/12 6:00	Su	0.00	1.00	NaN
5/12 12:00	Su	0.00	1.00	NaN
5/12 18:00	Su	0.27	0.75	NaN
5/12 24:00	Su	0.28	0.75	0.58
5/13 6:00	Mc	0.28	0.75	0.58
5/13 12:00	Mc	0.29	0.75	0.58
5/13 18:00	Mc	0.19	0.50	0.58
5/13 24:00	Mc	0.19	0.50	0.58
5/14 6:00	Tu	0.19	0.50	0.58
5/14 12:00	Tu	0.19	0.50	0.58
5/14 18:00	Tu	0.38	1.00	0.58
5/14 24:00	Tu	0.38	1.00	0.58
5/15 6:00	We	0.38	1.00	0.58
5/15 12:00	We	0.38	1.00	0.58
5/15 18:00	We	0.38	1.00	0.58
5/15 24:00	We	0.38	1.00	0.58
5/16 6:00	Th	0.38	1.00	0.58
5/16 12:00	Th	0.39	1.00	0.58
5/16 18:00	Th	0.39	1.00	0.58
5/16 24:00	Th	0.39	1.00	0.58
5/17 6:00	Fri	0.00	1.00	NaN
5/17 12:00	Fri	0.00	1.00	NaN
5/17 18:00	Fri	0.00	1.00	NaN

The current Regulated Drift Index is then used in conjunction with the average Pool Elevation over the current timestep and the Regulated Spill Index Table to obtain the current timestep's Regulated Spill.

If it is the first DRIFT timestep, a Gate Index must be calculated. This is done by using an average of the Pool Elevation at t-2 (when available) and at t-1, the previous regulated spill, the greatest and least spill values, and the Regulated Spill Index Table. If there is a non-zero capacity fraction, the value used for previous regulated spill is adjusted as follows:

$$\text{Regulated Spill}[t-1]_{Adj} = \frac{\text{Regulated Spill}[t-1] - \text{Closed Gate Overflow}[t-1]}{\text{Regulated Spill Capacity Fraction}[t-1]}$$

This computes the spill that would have occurred if all the gates were available. If the previous regulated spill is less than the smallest possible Regulated Spill or greater than the largest possible Regulated Spill (according to the Regulated Spill Index Table), a **RiverWare**™ error is flagged and the run is aborted.

The Regulated Drift Index slot is then set for the current timestep. Finally, the Regulated Spill can be determined from the three dimensional interpolation of the Regulated Spill Index Table using the average Pool Elevation over the timestep and the Regulated Spill Index.

At the end of the algorithm, the computed regulated spill (assuming all the gates are available) is multiplied by the Capacity Fraction [t] to determine spill that will occur with the given capacity fraction. This Regulated Spill is then set on the slot.

24.1.3.5 Regulated and Unregulated

The Regulated and Unregulated method models **Spill** through one controlled, **Regulated Spill**, and one uncontrolled spillway, **Unregulated Spill**. First, the **Unregulated Spill** can not be specified (input or via rules) or a **RiverWare™** error will abort the run. The **Unregulated Spill** is a function of the average reservoir **Pool Elevation** and takes precedence (i.e. water goes through it first) over other types of outflow (i.e. **Release** or **Turbine Release** or **Regulated Spill**) in the reservoir. When this method is chosen the user category Unregulated Spill Type , Section 24.1.4, will appear.

Second, the user may specify (input or via rules) either

- No slots
- **Spill** or
- **Regulated Spill**

If one is specified and there is excess flow which cannot be met by the (**Turbine**) **Release**, an error will be flagged and the simulation halted.

Slots Specific to this Method

☞ Regulated Spill

Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare™**. If regulated spill is set by the user and the value is greater than the required spill, an error is flagged and the simulation is halted.
Links: Usually not linked. But, it can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☞ Regulated Spill Capacity Fraction

Type: Series Slot
Units: DECIMAL

Description: The fraction of the Regulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.
I/O: Input, set by a rule, or output
Links: Not linkable

☞ REGULATED SPILL DRIFT INDEX

Type: Series Slot
Units: NONE
Description: gate setting index
Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Regulated Spill Index Table.
Links: Not linkable

☞ REGULATED SPILL INDEX TABLE

Type: Table Slot
Units: NOUNITS VS. LENGTH VS. FLOW
Description: Gate Index vs. Pool Elevation vs. Regulated Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.
Links: Not linkable

Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column 3.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130

Gate Index	Pool Elevation	Spill
4	550	180
4	600	230

☞ Regulated Spill Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding Max Regulated Spill values
Information:
I/O: Required input
Links: Not linkable

☞ Unregulated Spill

Type: Series Slot
Units: FLOW
Description: spill corresponding to the average Pool Elevation over the timestep
Information:
I/O: Output only
Links: Not linkable

☞ Unregulated Spill Capacity Fraction

Type: Series Slot
Units: Decimal
Description: The fraction of the Unregulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 50 ft of a 1000 ft long crest is blocked, the Unregulated Spill Capacity Fraction would be input to 0.95.
I/O: Input, set by a rule, or output
Links: Not linkable

☞ Unregulated Spill Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding Unregulated Spill values
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required input
Links: Not linkable

Note: Regulated Spill and Spill are both output slots if neither is input by the user. Only one of these slots, however, can be input for any timestep. When this is the case, the other slot will be an output slot.

The first step in the regulatedPlusUnregSpillCalc method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 24.1.5.2. If not, the minRegSpill is 0.0. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \\ \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

The Unregulated Spill is then calculated through the steps described in the “Unregulated”. If the Unregulated Spill is input by the user, a **RiverWare**™ error is flagged and the run is aborted.

Release is then checked to see if it has been calculated. If Release is not known, it means that the method is being called for the first time for the particular timestep and the following steps are taken:

1. If both the Spill and Regulated Spill are input (remember setting a DRIFT flag is considered an input), a **RiverWare**™ error is flagged and the run is aborted.

2. If Spill is input and it is greater than the sum of the Unregulated Spill and the maxRegSpill, a **RiverWare**™ error is flagged which states that the requested Spill cannot be met. If the Spill is input and less than the Unregulated Spill plus minRegSpill, a **RiverWare**™ error is flagged. Otherwise, Regulated Spill is calculated as Spill minus Unregulated Spill.

3. If Regulated Spill is input by the user and the DRIFT flag is set, a function is called to perform the drift calculations. A description of the DRIFT function is given in the Regulated Spill section. If Regulated Spill is input and greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted. Otherwise, Spill is set equal to the Regulated Spill plus Unregulated Spill.

4. If neither Regulated Spill nor Spill are input, Regulated Spill is set equal to the minRegSpill and Spill is set equal to Unregulated Spill plus minRegSpill.

After the Release has been calculated, the Regulated and Unregulated function may be called a second time if the sum of Release and Spill is less than the Outflow.

The following calculations are performed if the function is called for the second time:

1. If either Spill or Unregulated Spill are input, an error is posted because there are no free spill variables and **RiverWare**™ is unable to allocate the excess Outflow.

2. The Regulated Spill is calculated using the following equation:

$$\text{Regulated Spill} = \text{Outflow} - \text{Turbine Release} - \text{Unregulated Spill}$$

3. If Regulated Spill is greater than the maxRegSpill, a **RiverWare**™ error is posted stating that the Outflow is greater than the spillway capacities and Release and the run is aborted.

4. Spill is calculated as Regulated Spill plus Unregulated Spill.

24.1.3.6 Regulated and Bypass

The Regulated and Bypass method models spill through two regulated spillways called **Regulated Spill** and **Bypass**. The user may specify (input or via rules):

- No slots
- **Spill**
- **Spill** and **Bypass**
- **Spill** and **Regulated Spill**
- **Bypass**
- **Regulated Spill**, or
- **Bypass** and **Regulated Spill**

If all three slots are specified, an error will be issued. Also, if **Spill** is specified and there is excess flow that cannot be met by (**Turbine**) **Release**, a **RiverWare**™ error will be flagged and the simulation halted.

The order in which water goes through the various outflow structures depends on what is known. Input/Rules values take precedence, followed by (**Turbine**) **Release**, followed by **Regulated Spill**, and finally by **Bypass**. For example, on a timestep where there is zero (**Turbine**) **Release** and no spill slots are specified, outflows will first go through **Regulated Spill** and any flow greater than max regulated spill will go through **Bypass**.

Slots Specific to this Method

☞ **Bypass**

Type: Series Slot
Units: FLOW
Description: flow through the Bypass spillway
Information:

I/O: Optional; may be input by the user or determined by **RiverWare™**. If Bypass is set by the user and the value is greater than the required spill, a **RiverWare™** error is flagged and the simulation is halted.

Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Bypass behaves the same as if it were input.

☞ Bypass Capacity Fraction

Type: Series Slot
Units: Decimal
Description: The fraction of the Bypass structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 2 of 8 gates are unavailable, the Bypass Capacity Fraction would be input to 0.75.
I/O: Input, set by a rule, or output
Links: Not linkable

☞ Bypass Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index for the Bypass spillway
Information: If the user has set the DRIFT flag on the Bypass slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Bypass Index Table.
Links: Not linkable

☞ Bypass Index Table

Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Bypass Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Bypass spill slot, this data table must be provided.
Links: Not linkable

☞ Bypass Table

Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding maximum bypass spill values
Information:

I/O: Required input

Links: Not linkable

☞ REGULATED SPILL

Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare**™.

Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☞ REGULATED SPILL CAPACITY FRACTION

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Regulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.
I/O: Input, set by a rule, or output

Links: Not linkable

☞ REGULATED SPILL DRIFT INDEX

Type: Series Slot
Units: NONE
Description: gate setting index for the Regulated Spill
Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Regulated Spill Index Table.

Links: Not linkable

☞ REGULATED SPILL INDEX TABLE

Type: Table Slot
Units: NOUNITS VS. LENGTH VS. FLOW
Description: Gate Index vs. Pool Elevation vs. Regulated Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.

Links: Not linkable

Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column 3.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130
4	550	180
4	600	230

☞ Regulated Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding maximum regulated spill values
Information:
I/O: Required input
Links: Not linkable

Note: Spill, Regulated Spill, and Bypass can all be outputs if they are not specified by the user. The user may specify Spill, or either Regulated Spill or Bypass, or both Regulated Spill and Bypass. The slots which are not specified will be output slots.

The first step in the Regulated Spill method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 24.1.5.2. If not, the minRegSpill is 0.0. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

The next step is to obtain the maximum Bypass, maxBypass, by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Bypass Table. Then the Bypass Capacity Fraction is applied as follows:

$$\text{maxBypass} = \text{value from Bypass Table} \times \text{Bypass Capacity Fraction}$$

1. If Spill, Bypass, and Regulated Spill are all input/rules, a **RiverWare**™ Error is posted and the simulation run is aborted.
2. If Spill and Regulated Spill are input/rules by the user, the following steps are performed (Remember: If the DRIFT flag is set on the Regulated Spill slot, Regulated Spill is considered an input.):
   - Drift calculations are performed if the DRIFT flag is set on the Regulated Spill slot. See Regulated Spill for a description of the drift calculations.
   - If Regulated Spill is greater than either Spill or maxRegSpill a **RiverWare**™ error is flagged and the run is aborted.
   - If Regulated Spill is less than minRegSpill, a **RiverWare**™ error is flagged and the run is aborted.
   - Bypass is calculated as Spill minus Regulated Spill.
   - If Bypass is greater than the maxBypass, an error is flagged.
3. If Spill and Bypass are input, the following steps are taken (Remember: If the DRIFT flag is set on the Bypass slot, Bypass is considered an input.):
   - Drift calculations are performed if the DRIFT flag is set on the Bypass slot. The DRIFT calculations are performed in a similar manner to the drift calculations for Regulated Spill which are explained in the Regulated Spill method description.
   - If Bypass is greater than either Spill or the maxBypass, a **RiverWare**™ error is flagged and the simulation is aborted.
   - Regulated Spill is calculated as Spill minus Bypass.
   - If Regulated Spill is either greater than the maximum regulated spill or less than the minimum regulated spill, an error is flagged and the run is aborted.
4. If Spill is input but neither Regulated Spill nor Bypass are input or flagged as DRIFT, the following steps are taken:
   - Regulated Spill is set as the lesser value of either the Spill or the maxRegspill. The Regulated Spill also cannot be less than the minRegSpill.

- If Regulated Spill is less than Spill, Bypass is calculated as Spill minus Regulated Spill.
- If Bypass is greater than the maxBypass, a **RiverWare**™ error is flagged and the run is aborted.
- If Regulated Spill is equal to Spill, Bypass is set equal zero.

5. If Spill is not input, and both Bypass and Regulated Spill are input, the following steps are taken:
   - The drift calculations are performed for both Regulated Spill and Bypass if the DRIFT flags have been set. A description of the DRIFT calculations is contained in the Regulated Spill method.
   - Regulated Spill and Bypass are checked against the maxRegSpill and maxBypass, respectively. If either Regulated Spill is greater, a **RiverWare**™ error is flagged and the run is aborted.
   - Spill is calculated as Regulated Spill plus Bypass.
6. If the DRIFT flag is set on the Regulated Spill, the drift calculations are performed (as described in the Regulated Spill method) to calculate Regulated Spill. The calculated Regulated Spill value is then checked against spillway capacities. If only Regulated Spill is input, the value is checked against the spillway capacity. Spill is set equal to Regulated Spill and Bypass is set to zero if the Regulated Spill is less than or equal to the maxRegSpill. If the Regulated Spill is greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted and the simulation run is aborted.
7. If the DRIFT flag is set on the Bypass, the drift calculations are performed (as described in the Regulated Spill method) to calculate Bypass. The calculated Bypass value is then checked against spillway capacities. If only Bypass is input, the input value is checked against the spillway capacity. Spill is calculated as Bypass plus Regulated Spill. Regulated Spill is set to the minimum regulated spill, if the Bypass is less than the maximum bypass. If the Bypass is greater than the maxBypass, a **RiverWare**™ error is posted and the simulation run is aborted.
8. If no slots are input, Spill is set equal minRegSpill. Bypass is set to zero and Regulated Spill is set to minRegSpill.

After Release has been calculated, the Regulated and Bypass function may be called a second time if the sum of the Release and Spill is less than the Outflow.

The following calculations are performed if the function is called for the second time:

1. If either Spill is input, or both Regulated Spill and Bypass are input, a **RiverWare**™ error is flagged and the simulation run is aborted because there are no free spill variables.

2. If only Regulated Spill is input or flagged as DRIFT, Bypass is recalculated using the following formula:

$$\text{Bypass} = \text{Outflow} - \text{Regulated Spill} - \text{Turbine Release}$$

Bypass is then checked against its spillway capacity and a **RiverWare™** error is flagged and the simulation is aborted the spillway capacity is exceeded. Spill is calculated as Bypass plus Regulated Spill if the Bypass is less than or equal to maximum allowable bypass.

3. If only Bypass is input or flagged as DRIFT, Regulated Spill is recalculated using the following formula:

$$\text{Regulated Spill} = \text{Outflow} - \text{Bypass} - \text{Turbine Release}$$

The Regulated Spill is then checked against its spillway capacity and minRegSpill. A **RiverWare™** error is posted and the simulation run is aborted if the spillway capacity is exceeded or is less than minimum. Spill is calculated as the sum of Bypass and Regulated Spill.

4. If neither Bypass nor Regulated Spill is input, the following steps are performed:
   - A local variable, excess, is calculated as Outflow minus Turbine Release minus minRegSpill.
   - Regulated Spill is set equal to the lesser value of excess or maxRegSpill but must be greater than minRegSpill.
   - If Regulated Spill is less than the excess, Bypass is calculated as excess minus Regulated Spill.
   - Bypass is checked against its spillway capacity. If Bypass is greater than the maxBypass, a **RiverWare™** error is posted and the simulation run is aborted.
   - Spill is calculated as the Bypass plus the Regulated Spill.
   - If Regulated Spill is equal to the Excess, Bypass is set equal to zero and Spill is set equal to Regulated Spill.

24.1.3.7 Regulated, Bypass and Unregulated

This method models spill through two controlled spillways called **Bypass** and **Regulated Spill** and one uncontrolled spillway called **Unregulated Spill**. The user may not specify (input or via rules) the **Unregulated Spill**. This value is always output and is a function of the average reservoir Pool Elevation. The user may specify (input or rules):

- No slots
- **Spill**

- **Spill** and **Bypass**
- **Spill** and **Regulated Spill**
- **Bypass**
- **Regulated Spill**, or
- **Bypass** and **Regulated Spill**

If **Spill**, **Regulated Spill**, and **Bypass** are specified, an error will be issued. Also, if **Spill** is specified and there is excess flow that cannot be met by (**Turbine**) **Release**, a **RiverWare**™ error will be flagged and the simulation halted.

The order in which water goes through the various outflow structures depends on what is known. **Unregulated Spill** takes precedence, followed by input/rules values, followed by (**Turbine**) **Release**, followed by **Regulated Spill**, and finally by **Bypass**. For example, on a timestep where there is zero (**Turbine**) **Release** and no spill slots are specified, outflows will first go through **Unregulated Spill** (computed based on pool elevation), then **Regulated Spill** up to capacity and any excess flows will go through **Bypass**.

Slots Specific to this Method

☞ Bypass

Type: Series Slot
Units: FLOW
Description: flow through the Bypass spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare**™. If Bypass is set by the user and the value is greater than the required spill, a **RiverWare**™ error is flagged and the simulation is halted.
Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Bypass behaves the same as if it were input.

☞ Bypass Capacity Fraction

Type: Series Slot
Units: Decimal
Description: The fraction of the Bypass structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 2 of 8 gates are unavailable, the Bypass Capacity Fraction would be input to 0.75.
I/O: Input, set by a rule, or output
Links: Not linkable

☞ Bypass Drift Index

Type: Series Slot

Units: NONE
Description: gate setting index for the Bypass spillway
Information: If the user has set the DRIFT flag on the Bypass slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if not set by the user, the index is calculated from the Bypass Index Table.
Links: Not linkable

☞ Bypass Index Table
Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Bypass Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Bypass spill slot, this data table must be provided.
Links: Not linkable

☞ Bypass Table
Type: Table Slot
Units: LENGTH vs. FLOW
Description: Pool Elevation vs. corresponding maximum bypass spill values
Information:
I/O: Required input
Links: Not linkable

☞ Regulated Spill
Type: Series Slot
Units: FLOW
Description: flow through the regulated spillway
Information:
I/O: Optional; may be input by the user or determined by **RiverWare**™.
Links: Usually not linked. It can be linked to an expression slot if that expression slot fully evaluates at the beginning of timestep; in this case, Regulated Spill behaves the same as if it were input.

☞ Regulated Spill Capacity Fraction
Type: Series Slot
Units: DECIMAL
Description: The fraction of the Regulated Spill structure that is available.

Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be input to 0.875.
I/O: Input, set by a rule, or output
Links: Not linkable

☞ Regulated Spill Drift Index

Type: Series Slot
Units: NONE
Description: gate setting index for the Regulated Spill
Information: If the user has set the DRIFT flag on the Regulated Spill slot, the gate setting index from the previous timestep is maintained.
I/O: Optional; if this slot is not set by the user, the gate index is calculated from the Regulated Spill Index Table.
Links: Not linkable

☞ Regulated Spill Index Table

Type: Table Slot
Units: NOUNITS vs. LENGTH vs. FLOW
Description: Gate Index vs. Pool Elevation vs. Regulated Spill
Information:
I/O: Optional; if the user sets the DRIFT flag on the Regulated Spill slot, this data table must be provided.
Links: Not linkable

Data must be entered into the table in increasing blocks of the same Gate Index value for the 3-dimensional table interpolator to work correctly. For every block of same gate indices in column 1, Pool Elevations should be listed in increasing order in column 2, and the corresponding Spills in column 3.

Gate Index	Pool Elevation	Spill
2	500	110
2	550	160
2	600	210
3	500	120
3	550	170
3	600	220
4	500	130
4	550	180
4	600	230

☞ Regulated Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding maximum regulated spill values
Information:
I/O: Required input
Links: Not linkable

☞ Unregulated Spill

Type: Series Slot
Units: FLOW
Description: spill corresponding to the average Pool Elevation over the timestep
Information:
I/O: Output only
Links: Not linkable

☞ Unregulated Spill Capacity Fraction

Type: Series Slot
Units: DECIMAL
Description: The fraction of the Unregulated Spill structure that is available.
Information: If not input or set by the user, it defaults to 1.0. The value must be between 0.0 and 1.0, inclusive. Example: if 50 ft of a 1000 ft long crest is blocked, the Unregulated Spill Capacity Fraction would be input to 0.95.
I/O: Input, set by a rule, or output
Links: Not linkable

☞ Unregulated Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding unregulated spill values
Information:
I/O: Required input
Links: Not linkable

Note: Spill, Regulated Spill, and Bypass may be output slots if they are not specified as input by the user. The user may specify either Spill or Regulated Spill or Bypass, or both Unregulated Spill and Bypass as input. The slots which are not set as input will be output slots.

The first step in the Regulated Spill method is to obtain the minRegSpill. If the Closed Gate Overflow method is selected, the minRegSpill is computed as described Section 24.1.5.2. If not, the minRegSpill is 0.0. Next, the maximum regulated spill, maxRegSpill is computed by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Regulated Spill Table. Then the Regulated Spill Capacity Fraction and minRegSpill are applied as follows:

$$\text{maxRegSpill} = \text{value from Regulated Spill Table} \times \text{Regulated Spill Capacity Fraction} + \text{minRegSpill}$$

The next step is to obtain the maximum Bypass, maxBypass by looking up the average pool elevation (i.e. the average of the current Pool Elevation estimate and the previous Pool Elevation) on the Bypass Table. Then the Bypass Capacity Fraction is applied as follows:

$$\text{maxBypass} = \text{value from Bypass Table} \times \text{Bypass Capacity Fraction}$$

The Unregulated Spill is then calculated through the steps described in the “Unregulated”. If the Unregulated Spill is input by the user, a **RiverWare**™ error is flagged and the run is aborted.

If Spill, Bypass, and Regulated Spill are input, Spill is overdetermined and a **RiverWare**™ error is flagged and the simulation run is aborted.

If Release has not been calculated, the method is executing for the first time in the current timestep and the following steps are taken:

1. If Spill is input by the user and Unregulated Spill is greater than the Spill, a **RiverWare**™ error is flagged and the simulation run is aborted.
2. If Spill and Regulated Spill are input the following steps are taken (Remember: If the DRIFT flag is set on the Regulated Spill slot, Regulated Spill is considered an input):
    - Drift calculations are performed if the DRIFT flag is set on the Regulated Spill slot. See Regulated Spill for a description of the drift calculations.
    - If Regulated Spill is greater than either Spill or maxRegSpill, a **RiverWare**™ error is flagged and the run is aborted.
    - If Regulated Spill is less than minRegSpill, a **RiverWare**™ error is flagged and the run is aborted.
    - Bypass is calculated as Spill minus Regulated Spill minus Unregulated Spill.
    - If Bypass is either greater than the maxBypass or less than zero, an error is flagged.
3. If Spill and Bypass are input, the following steps are taken (Remember: If the DRIFT flag is set on the Bypass slot, Bypass is considered an input.):

- Drift calculations are performed if the DRIFT flag is set on the Bypass slot. The DRIFT calculations are performed in a similar manner to the drift calculations for Regulated Spill, which are explained in the Regulated Spill method description.
- If Bypass is greater than either Spill or the maxBypass, a **RiverWare**™ error is flagged and the run is aborted.
- Regulated Spill is calculated as Spill minus Bypass minus Unregulated Spill.
- If Regulated Spill is either greater than the maxRegSpill or less than the minRegSpill, an error is flagged and the run is aborted.

4. If Spill is input by the user and neither Bypass nor Regulated Spill are input, the following steps are taken:

- Regulated Spill is set as the lesser value of either the Spill minus Unregulated Spill or the maxRegSpill. The Regulated Spill also cannot be less than the minRegSpill.
- If Regulated Spill is less than Spill minus Unregulated Spill, Bypass is set equal to Spill minus Regulated Spill minus Unregulated Spill.
- If Bypass is greater than the maxBypass, a **RiverWare**™ error is flagged and the run is aborted.
- If Regulated Spill is equal to Spill minus Unregulated Spill, Bypass is set equal to zero.

5. If Spill is not input, but both Bypass and Regulated Spill are input, the following steps are taken:

- The drift calculations are performed for both Regulated Spill and Bypass if the DRIFT flags have been set. A description of the DRIFT calculations is contained in the Regulated Spill method.
- Regulated Spill and Bypass are checked against the maxRegSpill and maxBypass, respectively. If either Regulated Spill is greater, a **RiverWare**™ error is flagged and the run is aborted. If Regulated Spill is less than the minRegSpill, an error is issued.
- Spill is calculated as the sum of Regulated Spill, Bypass, and Unregulated Spill.

6. If the DRIFT flag is set on the Regulated Spill, the drift calculations are performed (as described in the Regulated Spill method) to calculate Regulated Spill. The calculated Regulated Spill Value is then checked against spillway capacities. If only Regulated Spill is input, the value is checked against the spillway capacity. Spill is calculated as Regulated Spill plus Unregulated Spill and Bypass is set to zero, if the Regulated Spill is less than the maxRegSpill. If the Regulated Spill is greater than the maxRegSpill or less than the minRegSpill, a **RiverWare**™ error is posted and the simulation run is aborted.

7. If the DRIFT flag is set on the Bypass, the drift calculations are performed (as discussed in the Regulated Spill method) to calculated the Bypass. The calculated Bypass value is then checked against spillway capacities. If only Bypass is input, the input value is checked against the spillway capacity. Spill is calculated as Bypass plus Unregulated Spill and Regulated Spill is set to the minRegSpill, if the Bypass is less than the maxBypass. If the Bypass is greater than the maxBypass, a **RiverWare**™ error is posted and the simulation run is aborted.

8. If no slots are input, Spill is set equal to Unregulated Spill plus minRegSpill. Bypass is set to zero and Regulated Spill is set to minRegSpill.

After Release has been calculated, the Regulated, Bypass and Unregulated function may be called a second time if the sum of the Release and Spill is less than the Outflow.

The following calculations are performed if the function is called for the second time:

1. If either Spill is input or both Regulated Spill and Bypass are input, a **RiverWare**™ error is flagged and the simulation run is aborted because there are no free spill variables.

2. If only Regulated Spill is input or flagged as DRIFT, Bypass is recalculated using the following formula:

$$\text{Bypass} = \text{Outflow} - \text{Regulated Spill} - \text{Unregulated Spill} - \text{Turbine Release}$$

The Bypass is then checked against its spillway capacity. A **RiverWare**™ error is posted and the simulation run is aborted if the spillway capacity is exceeded. Spill is calculated as the sum of Bypass, Unregulated Spill, and Regulated Spill if the Bypass is less than or equal to the maxBypass.

3. If only Bypass is input, Regulated Spill is recalculating the following formula:

$$\text{Regulated Spill} = \text{Outflow} - \text{Bypass} - \text{Unregulated Spill} - \text{Turbine Release}$$

The Regulated Spill is then checked against its spillway capacity and minRegSpill. A **RiverWare**™ error is posted and the simulation run is aborted if the spillway capacity is exceeded or is less than minimum. Spill is calculated as the sum of Bypass, Regulated Spill, and Unregulated Spill if the Regulated Spill is less than or equal to the maximum allowable regulated spill.

4. If neither Regulated Spill nor Bypass are input, the following steps are performed:
   - A local variable, excess, is calculated as Outflow minus Unregulated Spill minus Turbine Release minus minimum regulated spill.
   - Regulated Spill is set equal to the lesser value of excess or maxRegSpill but must be greater than minRegSpill.

- If Regulated Spill is less than excess, Bypass is calculated as excess minus Regulated Spill.
- Bypass is checked against its spillway capacity. If Bypass is greater than the maxBypass, a **RiverWare™** error is posted and the simulation run is aborted.
- Spill is set equal to the sum of Bypass, Regulated Spill, and Unregulated Spill.
- If Regulated Spill is equal to Excess, Bypass is set equal to zero and Spill is set equal to the sum of Regulated Spill and Unregulated Spill.

24.1.3.8 Bypass, Regulated and Unregulated

Note: This user method is the similar to the **Regulated, Bypass and Unregulated** method but switches the order of the **Bypass** and **Regulated Spill** outlet works. This method is preferable in institutional cases where the term "**Bypass**" is favored over the term "**Regulated Spill**". Other than the order reversal, the functionality is similar to the Regulated, Bypass and Unregulated method.

This method models spill through two controlled spillways called **Bypass** and **Regulated Spill** and one uncontrolled spillway called **Unregulated Spill**. The user may not specify (input or via rules) the **Unregulated Spill**. This value is always output and is a function of the average reservoir **Pool Elevation**. The user may specify (input or rules):

- No slots
- **Spill**
- **Spill** and **Bypass**
- **Spill** and **Regulated Spill**
- **Bypass**
- **Regulated Spill**, or
- **Bypass** and **Regulated Spill**

If **Spill**, **Regulated Spill**, and **Bypass** are specified, an error will be issued. Also, if **Spill** is specified and there is excess flow that cannot be met by (**Turbine**) **Release**, a **RiverWare™** error will be flagged and the simulation halted.

The order in which water will go through the various outflow structures depends on what is known. **Unregulated Spill** takes precedence, followed by input/rules values, followed by (**Turbine**) **Release**, followed by **Bypass**, and finally by **Regulated Spill**. For example, on a timestep where there is zero (**Turbine**) **Release** and no spill slots are specified, outflows will first go through **Unregulated Spill** (required based on pool elevation), then **Bypass** up to capacity and any excess flows will go through **Regulated Spill**.

Please see the **regPlusBypassPlusUnregSpill** method for a description of the slots particular to this method and the algorithm of this method. The algorithm for this method is only different in that **Bypass** takes precedence over **Regulated Spill**.

24.1.4 Unregulated Spill Type

This category is only visible when a method using **Unregulated Spill** is chosen. The three Unregulated Spill Types are Bare Crest Only, Two Unregulated Flow, and Three Unregulated Flows.

24.1.4.1 Bare Crest Only

The Bare Crest Only method is the default method in the Unregulated Spill Type Category. The method assumes an unobstructed spillway where the flow over the spillway is a function of the **Unregulated Spill Table**. There are no slots specifically associated with this method.

24.1.4.2 Two Unregulated Flows

When the **Two Unregulated Flows** method is selected, flow over the spillway is a function of the **Unreg Flow 2 Spill Table.** If the pool elevation meets or exceeds the **Unreg Flow 2 Failure Elevation**, flow becomes a function of the **Unregulated Spill Table.**

Note: This method originally was called **Flashboards** but was renamed to be more general. (Flashboards are wooden boards installed in the unregulated spillway so that the reservoir may store more water than what the spillways themselves would allow.)

☞ Unreg Flow 2 Avail and Failure Time

Type: Agg Series Slot
Units: FRACTION
Description: Availability of Unreg Flow 2 Spill Table, fraction of timestep when Unreg Flow 2 Spill Table is in use.
Information:
I/O: Optional: Availability may be input by user or set by a rule. Failure time is output only
Links: Not linkable

☞ Unreg Flow 2 Failure Elevation

Type: Table Slot
Units: LENGTH
Description: Pool Elevation at which Unreg Flow 2 Spill Table is no longer used.
Information:
I/O: Required Input
Links: Not linkable

☞ Unreg Flow 2 Spill Table

Type: Table Slot

Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding unregulated flow 2 spill values.
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required Input
Links: Not linkable

METHOD DETAILS

To determine the spill tables to use during a timestep based on availability of Unreg Flow 2, the following three possibilities are checked:

1. Availability is input or set by a rule: no change from Unreg Flow 2 to Unregulated Spill over timestep, calculate spill based on availability.

2. Availability is 0: no change from Unreg Flow 2 to Unregulated Spill, calculate spill based on Unregulated Spill Table.

3. Availability is greater than 0, check if failure from Unreg Flow 2 to Unregulated Spill occurs during the timestep.

During the third case, the spill over the timestep is calculated using the respective table and multiplied by the availability to find the total spill over the timestep. The total spill is used to predict the pool elevation at the end of the timestep. The calculated pool elevation is compared to the Unreg Flow 2 Failure Elevation. If a failure from Unreg Flow 2 to Unregulated Spill is found to occur any time during the timestep, the failure time is recorded and the ending pool elevation is re-calculated to account for the change in spill due to the change from Unreg Flow 2 to Unregulated Spill during the timestep.

Note: If a failure from Unreg Flow 2 to Unregulated Spill occurs during a dispatch, the first time of this failure is used for the remainder of the dispatch.

The time of failure during the timestep is used to determine what portion of the timestep needs interpolation from each of the two spill tables.

24.1.4.3 Three Unregulated Flows

When the **Three Unregulated Flows** is selected, flow over the spillway is a function of the **Unreg Flow 3 Spill Table.** If the pool elevation meets or exceeds the **Unreg Flow 3 Failure Elevation**, flow becomes a function of the **Unreg Flow 2 Spill Table.** If the pool elevation meets or exceeds the **Unreg Flow 2 Failure Elevation**, flow becomes a function of the **Unreg Flow Spill Table.** To summarize, Unreg Flow 3 fails first, then Unreg Flow 2 fails next as the pool rises. Therefore, the Unreg Flow 2 Failure Elevation should be higher than the Unreg Flow 3 Failure Elevation.

Note: This method was originally called the **Flashboards and Superboards** method but was renamed to be more general. Flashboards and superboards are wooden boards installed in the unregulated spillway so that the reservoir may store more water than what the spillways themselves would allow. The superboards can only be installed if the flashboards are in place.

☞ Unreg Flow Avail and Failure Time

Type: Agg Series Slot
Units: FRACTION
Description: Availability and failure time of Unreg flow 2 and 3 Spill tables.
Information:
I/O: Optional: Availability may be input by user or set by a rule. Failure time is output only
Links: Not linkable

☞ Unreg Flow 3 Failure Elevation

Type: Table Slot
Units: LENGTH
Description: Pool Elevation at which Unreg Flow 3 Spill Table is no longer used.
Information: The value in this slot should be lower than the value in the Unreg Flow 2 Failure Elevation.
I/O: Required Input
Links: Not linkable

☞ Unreg Flow 3 Spill Table

Type: Table Slot
Units: LENGTH VS. FLOW
Description: Pool Elevation vs. corresponding unregulated flow 3 spill values.
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required Input
Links: Not linkable

☞ Unreg Flow 2 Failure Elevation

Type: Table Slot
Units: LENGTH
Description: Pool Elevation at which Unreg Flow 2 Spill Table is no longer used.
Information: The value in this slot should be higher than the value in the Unreg Flow 2 Failure Elevation.
I/O: Required Input
Links: Not linkable

☞ Unreg Flow 2 Spill Table

Type: Table Slot
Units: Length vs. Flow
Description: Pool Elevation vs. corresponding unregulated flow 2 spill values.
Information: Must contain a row which corresponds to a spill of zero for interpolation purposes.
I/O: Required Input
Links: Not linkable

Method Details

To determine the spill tables to use during a timestep based on availability of Unreg Spill 2 and Unreg Spill 3, the following four possibilities are checked:

1. Both availabilities are input or set by a rule: no change in unregulated spill type over timestep, calculate spill based on availability.
2. One availability is input or set by a rule: Error, both or none must be input.
3. Availabilities are 0. No change in unregulated spill type over timestep. Calculate spill based on Unregulated Spill Table.
4. At least one availability is greater than 0, check for failure in unregulated spill type.

During the fourth case the spill over the timestep is calculated using the respective table and multiplied by the availability to find the total spill over the timestep. The total spill is used to project the pool elevation at the end of the timestep. The calculated pool elevation is compared to the failure elevations of Unreg Flow 2 or Unreg Flow 3. If failure in unregulated spill type is found to occur any time during the timestep, this failure time is recorded and the ending pool elevation is recalculated to account for the change in spill due to the change in unregulated spill type during the timestep.

Note: If a change in unregulated spill type occurs during a dispatch, the first time of this change is used for the remainder of the dispatch.

The time at which change in unregulated spill type occurred during the timestep is used to determine what portion of the timestep needs interpolation from each of the three spill tables.

24.1.5 Regulated Spill Overflow

The category, **Regulated Spill Overflow,** is added if one of the following "regulated" spill methods is selected:

- **Regulated Spill**
- **Regulated and Unregulated**
- **Regulated and Bypass**
- **Regulated, Bypass and Unregulated**
- **Bypass, Regulated and Unregulated**

24.1.5.1 None

This is the default, no-action method.

24.1.5.2 Closed Gate Overflow

This method models the uncontrolled flow over a closed regulated spill gate. This functionality uses the **Regulated Spill Capacity Fraction** to compute the default amount of spillway that is overtopped.

This functionality only applies to **Regulated Spill**, not **Bypass**.

THE FOLLOWING SLOTS WILL BE ADDED:

☞ **CLOSED GATE OVERFLOW**

Type: Series Slot
Units: FLOW
Description: Uncontrolled portion of the Regulated spill that overtops the gates.
Information: This value is computed by the regulated spill method as the value found on the **Closed Gate Overflow Table** multiplied by the **Closed Gate Overflow Capacity Fraction**
I/O: Output only
Links: Not linkable

☞ **CLOSED GATE OVERFLOW TABLE**

Type: Table Slot
Units: LENGTH VS FLOW
Description: Pool Elevation vs unregulated flow.
Information: This table is used to specify the rating curve for uncontrolled flow over the closed Regulated Spill gates. The values should be input to the table as though every Regulated Spill gate is closed. The tables would start with zero flow at or just below the top of the closed gates.
I/O: Input Only

Links: Not available

☞ Closed Gate Overflow Capacity Fraction

Type: Series Slot
Units: FRACTION
Description: The fraction of the closed gate overflow that is available.
Information: The value must be between 0.0 and 1.0, inclusive. If not input or set by a rule, it defaults to (1 - **Regulated Spill Capacity Fraction**). Example: if 1 of 8 gates are unavailable, the Regulated Spill Capacity Fraction would be set to 0.875 and the Closed Gate Overflow Capacity Fraction would default to 0.125.
I/O: Input, set by a rule, or output
Links: Not linkable

Method Details

If not input or set by a rule, **Closed Gate Overflow Capacity Fraction** defaults to (1 - **Regulated Spill Capacity Fraction**). This default indicates that the overflow only happens over gates that are closed. Otherwise, the user can specify the **Closed Gate Overflow Capacity Fraction** slot to say how much of the overflow structure is available.

When the reservoir is below the top of the gates, there is no **Closed Gate Overflow**. But once the reservoir is above the top of one or more closed gates, there is **Closed Gate Overflow**. The computation of this overflow is similar to the unregulated spill computation:

1. A temporary variable called "initHW" is created to represent the Pool Elevation at the beginning of the timestep. Likewise, "endHW" is created to represent the Pool Elevation at the end of the timestep (If the Pool Elevation at the end of the timestep is not known, endHW is set equal to the Pool Elevation at the beginning of the timestep.)

2. The "Closed Gate Overflow Crest" is found from the Closed Gate Overflow Table. It is the Pool Elevation that corresponds to an overflow of zero.

3. If both initHW and endHW are less than or equal to the Closed Gate Overflow Crest, Closed Gate Overflow is set equal to zero.

4. If both initHW and endHW are greater than the Closed Gate Overflow Crest, the average Pool Elevation is used to determine the Closed Gate Overflow from the Closed Gate Overflow Table.

$$\text{Closed Gate Overflow} = \text{Value from table} \times \text{Closed Gate Overflow Capacity Fraction}$$

5. If either initHW or endHW is greater than the Closed Gate Overflow Crest and the other is lower than the crest, the following evaluations and computations are performed:

$$\text{maxHW} = \text{the greater of initHW and endHW}$$

$$\text{minHW} = \text{the lesser of initHW and endHW}$$

$$\text{avgHW} = \frac{\text{maxHW} + \text{Closed Gate Overflow Crest}}{2}$$

$$\text{overflow fraction} = \frac{\text{maxHW} - \text{Closed Gate Overflow Crest}}{\text{maxHW} - \text{minHW}}$$

where:

maxHW = the maximum value of Pool Elevation over the timestep.

minHW = the minimum value of Pool Elevation over the timestep.

avgHW = the average Pool Elevation causing overflow over the timestep.

overflow fraction = corresponds to the fraction of the timestep during which overflow occurs.

A temporary variable called “temp overflow” is obtained from the linear interpolation of the Closed Gate Overflow Table using avgHW. Closed Gate Overflow is then calculated as:

$$\text{Closed Gate Overflow} = \text{overflow fraction} \times \text{temp overflow} \times \text{Closed Gate Overflow Capacity Fraction}$$

When allocating spills to various structures, the **Closed Gate Overflow** must occur at the same time as unregulated spills (i.e. before regulated or bypass). Then any remaining outflow can go through the regulated and or bypass spill structures. Therefore, the minimum regulated spill is computed as follows:

$$\text{Min Regulated Spill} = \text{value from Closed Gate Overflow Table} \times \text{Closed Gate Overflow Capacity Fraction}$$

Also, the **Closed Gate Overflow** is set equal to the Min Regulated Spill.

The functionality assumes that water is either flowing through the gate or over topping it, but not both. The method assumes that the Gate Overflow Table is fixed, that is the elevations in the table do not change. Thus if you had a gate stuck with 1/2 ft open at the bottom and there was still water going over the top, the table (which assumes the gate is closed) would be an incorrect rating.

24.1.6 Input Outflow Adjustment

This method category is only available if a method is selected in the Spill Calculation category. Its purpose is to adjust input Outflow values if they violate a physical constraint.

24.1.6.1 None

This is the default method. It performs no calculations and there are no slots associated with it. The Outflow values will not be adjusted if this method is selected.

24.1.6.2 Reduce Input Outflow

This method is used to reduce the input Outflow value whenever it exceeds the maximum reservoir outflow (due to outlet works capacity).

Slots Specific to this Method

☞ **Requested Outflow**

Type: Series Slot
Units: FLOW
Description: The Outflow value before being adjusted
Information: This slot is available so that the user can see when an Outflow value is adjusted. The value in this slot is the outflow value before being adjusted. A value exists in this slot only if the Outflow value is adjusted.
I/O: Output only
Links: Not linkable

If the Outflow slot value is greater than the maximum reservoir outflow, this method saves the Outflow value in the Requested Outflow slot. Then, the Maximum Capacity flag is set on the Outflow slot. The reservoir is then forced to re-dispatch with the Outflow set to Max Capacity (instead of the original, input value). When the reservoir solves the second time, it computes the maximum reservoir outflow and sets this value on the Outflow slot. The Maximum Capacity flag remains on the Outflow slot for the timestep in question (and will be saved with the model file).

24.1.6.3 Allow Excess Specified Outflows

This method allows input Outflows that exceed the maximum reservoir outflow (due to outlet capacity). Because the excess is above the maximum possible for the (Turbine) Release and Spill slots, it will not be classified as either. Instead, the excess is stored on a separate series slot for reporting or tracking.

Slots Associated With this Method:

☞ Outflow Exceeding Max

Type: Series Slot
Units: Flow
Description: The portion of the input Outflow that exceeds the sum of the Spill and (Turbine) Release.
Information: This slot tracks the amount that does not fit through the Release and Spill structures.
I/O: Output Only
Links: Not Linkable

Method Details:

Toward the end of each dispatch method, if Outflow is greater than the sum of Spill and (Turbine) Release, the Spill method is executed again to redistribute the Outflows to the appropriate spill structures. Within the Spill method, if there is still **no** room for the specified Outflow, the selected method in the **Input Outflow Adjustment** category is executed. When the **Allow Excess Specified Outflows** method is selected, it does the following:

If the Outflow does not have an input flag (I or Z), then the method exits and issues an error that there are excess outflows.

If the Outflow is input (I or Z flag), the method computes the difference between the specified Outflow and the maximum Outflow (i.e. Turbine Release + Spill). This excess outflow is then set on the **Outflow Exceeding Max** slot.

$$\text{Outflow Exceeding Max}[\,] = \text{Outflow}[\,] - ((\text{Turbine})\ \text{Release}[\,] + \text{Spill}[\,])$$

The method then exits successfully and returns to the Spill method and then the dispatch method. The dispatch method sets the spill and mass balance slots.

24.1.7 Future Value

The methods in this category are used to determine the future value of the energy that would have been generated by the water that was lost through the spillway.

24.1.7.1 None

None is the default method for the Future Value category. No calculations are performed by this method. There are no slots specifically associated with this method.

24.1.7.2 Cumulative Storage Value Table

SLOTS SPECIFIC TO THIS METHOD

☞ **MARGINAL STORAGE VALUE TABLE**

Type: Table
Units: VOLUME VS. $PER ENERGY
Description: Storage versus marginal value per unit energy
Information: This table should be increasing in storage, and usually decreasing in marginal value
I/O: Required input
Links: Not linkable

☞ **SPILL COST**

Type: Series Slot
Units: $
Description: Future cost of energy lost due to spilled water
Information:
I/O: Output only
Links: may be linked to the Spill Cost slot on the Thermal Object.

☞ **ANTICIPATED STORAGE**

Type: Series
Units: VOLUME
Description: The combination of Storage in the reservoir at the given timestep plus any flow (converted volume) that is in transit to the reservoir
Information: This slot represents the storage including any lagged flows that are already in a linked upstream reach, which will reach the reservoir at a later timestep. If there are no lagged reaches between this reservoir and the next upstream reservoir, Anticipated Storage will equal Storage. It is this storage value that will be used to calculate Cumulative Storage Value.
I/O: Output only

Links: May be linked

☞ Cumulative Storage Value

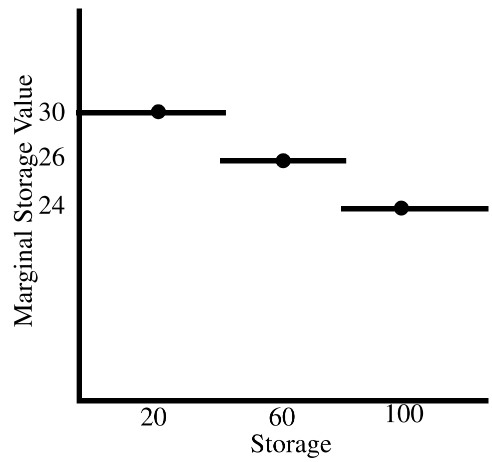
Type: Series Slot
Units: $
Description: Represents the future energy value of the current Anticipated Storage
Information:
I/O: Output only
Links: May be linked to the Total Cumulative Storage Value Slot on the Thermal Object

☞ Cumulative Storage Value Table

Type: Table
Units: VOLUME VS. $
Description: Anticipated Storage and cumulative value used to calculate the Cumulative Storage Value as a function of Anticipated Storage
Information: This table should be increasing in storage and usually increasing in cumulative storage value.
I/O: Required Input either by the user or automatically generated by **RiverWare**™ if the Cumulative Storage Value Table Automation method is selected.
Links: Not linkable

This method uses the Marginal Storage Value Table and the calculated Spill and Turbine Release to compute the Spill Cost. These are the only calculations performed by this method.

The correct marginal value is found from the current storage in the reservoir. If the current storage is than the midpoint between the first and second storage table values, the first marginal value is used. The second marginal value is used for a current storage above that midpoint to the midpoint between the second and third storage table values. The last marginal value is used for any current storage above the midpoint between the second-to-last and the last storage table value. An example is shown in Table 24 on page 1355



Storage	Marginal Value
20	30
60	26
100	24

Table: 24 Marginal Value Table

Assume that the current storage is 39. Therefore, this method would use 30 as the marginal value for use in the next computation. Assume that the current storage is 41. Therefore, this method would use 26 as the marginal value for use in the next computation.

Use of a table in this fashion is unique to this method.

Spill Cost is computed by the following equation:

$$\text{Spill Cost} = \text{Spill} \times \text{Marginal Storage Value} \times \text{Timestep Length}$$

The Cumulative Storage Value computation begins by first calculating Anticipated Storage. This is the sum of the reservoir Storage plus any flow already in transit to the reservoir in an upstream lagged reach. For example, assume a reservoir's Inflow slot is linked to a reach with a 3-hour lag time. In an hourly run, the reservoir's Anticipated Storage would be calculated as:

$$\text{Reservoir.Anticipated Storage} = \text{Reservoir.Storage} \\ + (\text{Reach.Inflow(-2)} + \text{Reach.Inflow(-1)} + \text{Reach.Inflow}) \times TimestepLength$$

If there are no lagged reaches between the reservoir and the next upstream reservoir, then Anticipated Storage will simply equal Storage.

The Cumulative Storage Value is then computed by interpolating from the Cumulative Storage Value Table using the calculated Anticipated Storage value.

24.1.8 Cumulative Storage Value Table Automation

This category allows the **RiverWare**™ simulation to automate the creation of the Cumulative Storage Value Table. This category is only visible if Cumulative Storage Value Table is selected in the Future Value category.

24.1.8.1 None

If this method is selected, no automation will be performed and the user must enter the data into the Cumulative Storage Value Table.

24.1.8.2 Marginal Value to Table

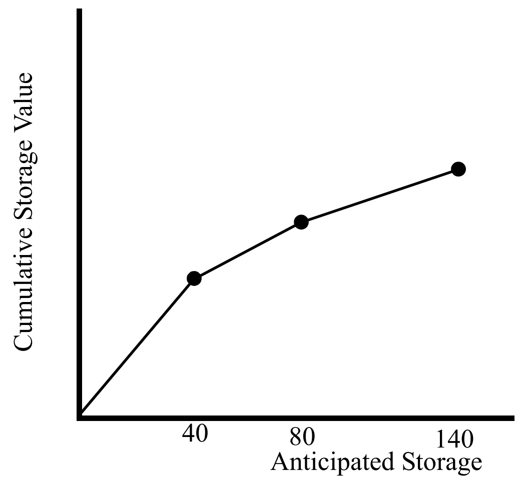
If this method is selected, the Marginal Storage Value table will be used as the source for the generation of the Cumulative Storage Value Table. This is the only calculation associated with this method. There are no slots associated specifically with this method.

This method uses information from the simulation slot Marginal Storage Value Table to generate a cumulative storage value table. The cumulative storage value can be thought of as the summation of the marginal storage values from a storage of 0 to the current storage. Therefore, the automation method finds the same midpoint values used by the simulation Future Value Calc method, and uses those points in the table.

If only one value exists in the Marginal Storage Value Table, then only two entries will exist in the Cumulative Storage Value Table. The two entries will be 0, and midway between the value in the Marginal Storage Value Table, and the maximum value set on the Storage slot. If more than one value exists, three or more points will result. An example is shown below of a Marginal Value Table and the resulting Cumulative Storage Value Table with a graph of the Cumulative Storage Value data.

Storage	Marginal Value
20	30
60	26
100	24

Table: 25 Marginal Value Table



Anticipated Storage	Cumulative Value
40	1200
80	2240
140	3680

Table: 26 Cumulative Storage Value Table

24.1.9 Hydrologic Inflow

The Hydrologic Inflow category allows **RiverWare**™ to accommodate inflows to a Reservoir that are not part of the main channel and/or are not gauged. The user methods in this category may be used to initialize the Hydrologic Inflow slot if it is required by the user. If the Hydrologic Inflow slot has been initialized, it is figured into the mass balance when the object dispatches.

24.1.9.1 None

None is the default method for the Hydrologic Inflow category. No calculations are performed by this method. There are no slots specifically associated with this method. If this method is selected, the Hydrologic Inflow slot is not initialized so it is no included in the mass balance.

24.1.9.2 Solve Hydrologic Inflow

The Solve Hydrologic Inflow method should be used when the user wishes **RiverWare**™ to calculate Hydrologic Inflow. Hydrologic Inflow will be solved for when the Reservoir dispatches. **RiverWare**™ will not overwrite any user input values for Hydrologic Inflow. Hydrologic Inflow is only solved for when a value is not input.

Slots Specific to this Method

☞ **Hydrologic Inflow**

Type: Series
Units: FLOW
Description: flow into the reservoir that is not gauged and/or does not enter through the main channel.
Information:
I/O: Optional; calculated if not input.
Links: Usually input or calculated but could be linked to the Outflow of any object or any other series slot.

☞ **Hydrologic Inflow Adjust**

Type: Series
Units: FLOW
Description: optional adjustment that can be made to the calculated Hydrologic Inflow
Information:
I/O: Optional; set to zero if not input by the user
Links: Not linkable

☞ **HYDROLOGIC INFLOW NET**

Type: Series
Units: FLOW
Description: sum of Hydrologic Inflow and Hydrologic Inflow Adjust
Information:
I/O: Output only
Links: Not linkable

The following steps are performed in the Solve Hydrologic Inflow method.

First, if Hydrologic Inflow Adjust is not set by the user, it is set equal to zero by **RiverWare**™. Then if Hydrologic Inflow is not set by the user, it is calculated in the dispatch method (see Dispatch Methods for detailed explanation). Hydrologic Inflow Net is calculated as the sum of Hydrologic Inflow and Hydrologic Inflow Adjust.

24.1.9.3 Input Hydrologic Inflow

The Input Hydrologic Inflow method should be used when the user wishes either to input the values of Hydrologic Inflow or have the values default to zero. **RiverWare**™ will not overwrite any user input values.

SLOTS SPECIFIC TO THIS METHOD

☞ **HYDROLOGIC INFLOW**

Type: Series
Units: FLOW
Description: flow into the reservoir that is not gauged and/or does not enter through the main channel.
Information:
I/O: Optional; defaults to zero if not input.
Links: Usually input or calculated but could be linked to the Outflow of any object or any other series slot.

☞ **HYDROLOGIC INFLOW ADJUST**

Type: Series
Units: FLOW
Description: optional adjustment that can be made to the calculated Hydrologic Inflow
Information:
I/O: Optional; set to zero if not input by the user.
Links: Not linkable

☞ **HYDROLOGIC INFLOW NET**

Type: Series
Units: FLOW
Description: sum of hydrologic Inflow and Hydrologic Inflow Adjust
Information:
I/O: Output only
Links: Not linkable

The algorithm used for this method is very simple. If Hydrologic Inflow is not input by the user, it is set equal to zero. Hydrologic Inflow Net is calculated as the sum of Hydrologic Inflow and Hydrologic Inflow Adjust.

24.1.9.4 Hydrologic Inflow and Loss

The Hydrologic Inflow and Loss method should be used when the user wishes to have negative inflows taken into account as unidentified losses.

SLOTS SPECIFIC TO THIS METHOD

☞ **HYDROLOGIC INFLOW**

Type: Series
Units: FLOW
Description: flow into the reservoir that is not gauged and/or does not enter through the main channel.
Information:
I/O: Optional; defaults to zero if not input.
Links: Usually input or calculated but could be linked to the Outflow of any object or any other series slot.

☞ **HYDROLOGIC INFLOW ADJUST**

Type: Series
Units: FLOW
Description: optional adjustment that can be made to the calculated Hydrologic Inflow
Information:
I/O: Optional; the Hydrologic Inflow Adjust may either be input by the user or set to zero if it is not input. If the Inflow is negative, it is added to Hydrologic Inflow Adjust.
Links: Not linkable

☞ **HYDROLOGIC INFLOW NET**

Type: Series
Units: FLOW

Description: sum of Hydrologic Inflow and Hydrologic Inflow Adjust
Information:
I/O: Output only
Links: Not linkable

This method calculates Hydrologic Inflow using the following process.

If the user has selected this method and negative inflows occur, this method will be called from the dispatch method. The Inflow (which in this case is a negative number) will be added to the Hydrologic Inflow Adjust. Then the Inflow is set equal to zero. Otherwise, this method behaves like the Input Hydrologic Inflow method.

24.1.9.5 Forecast Hydrologic Inflow

This method is used to forecast the hydrologic inflow based on known inflow values. When this method is selected, the Generate Forecast Hydrology category becomes visible. Within this category, methods are available to generate the hydrologic inflow forecast.

24.1.10 Generate Forecast Hydrology

This category contains methods that forecast the hydrologic inflow based on known inflow values. The user inputs the historical inflows to the reservoir and the methods adjust those values to represent a forecast. The methods in this category execute at the beginning of each timestep.

24.1.10.1 None

This method is the default for this category. It will result in an error if it is selected and a run is started.

24.1.10.2 Geometric Recession

On each timestep in the forecast period, this method will adjust the inflow hydrographs. If the Forecast Period method is selected for the Incremental Hydrologic Inflows on Subbasin category, the Cumulative Hydrologic Inflow is used to forecast and set the Forecasted Cumulative Hydrologic Inflow. If the Forecast Period methods is not selected, the input Deterministic Incremental Hydrologic Inflow slot is used to forecast and set the Hydrologic Inflow Forecast series slot.

SLOTS SPECIFIC TO THIS METHOD

☞ **HYDROLOGIC INFLOW FORECAST**

Type: Series
Units: FLOW
Description: The forecasted hydrologic inflow values
Information:
I/O: Output only
Links: Not linkable

☞ **DETERMINISTIC INCREMENTAL HYDROLOGIC INFLOW**

Type: Series
Units: FLOW
Description: This slots holds a timeseries of the actual hydrologic inflows to the reach. These values are then adjusted by the forecast method and set on the Hydrologic Inflow Forecast slot.
Information: At the end of the run, the Hydrologic Inflow Forecast slot will be identical to this slot. If the Full Run method is selected, values from the Incremental Hydrologic Inflow slot will be copied into this slot prior to the forecast. This slot is not used (i.e. inputs are ignored) if the Forecast Period method is selected.
I/O: Input or set to the values in the Incremental Hydrologic Inflow slot

Links: Not linkable

☞ Period of Perfect Knowledge

Type: Scalar
Units: FLOW
Description: Number of timesteps for which the forecast will equal the Deterministic Incremental Hydrologic Inflow, i.e., the forecast is known with complete certainty.
Information: Minimum value of 1; maximum value equal to the number of timesteps in the forecast period.
I/O: Input only
Type: Not linkable

☞ Recession Factor

Type: Scalar
Units: NONE
Description: A decimal value that is multiplied by the previous Hydrologic Inflow Forecast value to determine the current value after the Period of Perfect Knowledge.
Information:
I/O: Input only
Links: Not linkable

This method uses different slots and sets different slots if the Forecast Period is selected for the Incremental Hydrologic Inflows on Subbasin category. But, there is always a Source slot and a Target slot where the Source slot is input and used to forecasted and set the Target slot. If the Forecast Period is selected, the computational subbasin calls this method and uses the Cumulative Hydrologic Infow (Source) to forecast and set the Forecasted Cumulative Hydrologic Inflow (Target). If it is not selected, the reservoir calls this method at the beginning of the timestep and uses the Deterministic Incremental Hydrologic Inflow (Source) slot to forecast and set the Hydrologic Inflow Forecast slot (Target). The following uses the Source/Target terminology to describe the methods.

The Source slot values are required inputs for each timestep. At the beginning of each controller timestep, the Geometric Recession method is executed. For each forecast timestep within the period of perfect knowledge, the Target is set to the Source value. For each forecast timestep after the period of perfect knowledge, the Target is set by multiplying the value of the Target from the previous timestep by the constant recession factor.

A value for the Source slot must be known at every timestep during the run. If Target values are desired past the end of the run, there must also be values in the Source slot at timesteps past the end of the run. If values for Source slot are not entered past the end of the run, the Target values for these timesteps are assumed to be zero.

24.1.10.3 Exponential Recession

On each timestep in the forecast period, this method will adjust the inflow hydrographs. If the Forecast Period method is selected for the Incremental Hydrologic Inflows on Subbasin category, the Cumulative Hydrologic Inflow is used to forecast and set the Forecasted Cumulative Hydrologic Inflow. If the Forecast Period methods is not selected, the input Deterministic Incremental Hydrologic Inflow slot is used to forecast and set the Hydrologic Inflow Forecast series slot.

SLOTS SPECIFIC TO THIS METHOD

☞ **HYDROLOGIC INFLOW FORECAST**

Type: Series
Units: FLOW
Description: The forecasted inflow values computed from the Deterministic Incremental Hydrologic Inflow
Information:
I/O: Output only
Links: Not linkable

☞ **DETERMINISTIC INCREMENTAL HYDROLOGIC INFLOW**

Type: Series
Units: FLOW
Description: This slots holds a timeseries of the actual hydrologic inflows to the reach. These values are then adjusted by the forecast method and set on the Hydrologic Inflow Forecast slot.
Information: At the end of the run, the Hydrologic Inflow Forecast slot will be identical to this slot. If the Full Run method is selected, values from the Incremental Hydrologic Inflow slot will be copied into this slot prior to the forecast. This slot is not used (i.e. inputs are ignored) if the Forecast Period method is selected.
I/O: Input or set to the values in the Incremental Hydrologic Inflow slot
Links: Not linkable

☞ **FORECAST PERIOD**

Type: Table
Units: NONE
Description: Number of timesteps, not including the current timestep, that the inflow hydrograph will be adjusted.
Information:
I/O: Input only
Links: Not linkable

☞ Period of Perfect Knowledge

Type: Scalar
Units: FLOW
Description: Number of timesteps for which the forecast will equal the Deterministic Incremental Hydrologic Inflow, i.e., the forecast is known with complete certainty.
Information: Minimum value of 1; maximum value equal to the number of timesteps in the forecast period.
I/O: Input only
Links: Not linkable

☞ Minimum Forecasted Flow

Type: Series
Units: FLOW
Description: The minimum forecasted flow.
Information: If the computed value for Hydrologic Inflow Forecast is less than the Minimum Forecasted Flow, it is set to the Minimum Forecasted Flow.
I/O: Input only
Links: Not linkable

☞ Low Flow Threshold

Type: Scalar
Units: FLOW
Description: The flow rate that dictates whether to use the Low Flow Recession Coefficient or the High Flow Recession Coefficient.
Information:
I/O: Input only
Links: Not linkable

☞ Low Flow Recession Coefficient

Type: Scalar
Units: NONE
Description: The recession coefficient used when the Deterministic Incremental Hydrologic Inflow (at the end of the Period of Perfect Knowledge) is below or equal to the Low Flow Threshold.
Information:
I/O: Input only
Links: Not linkable

☞ High Flow Recession Coefficient

Type: Scalar

Units: NONE
Description: The recession coefficient used when the Deterministic Incremental Hydrologic Inflow (at the end of the Period of Perfect Knowledge) is above the Low Flow Threshold.
Information:
I/O: Input only
Links: Not linkable

This method uses different slots and sets different slots if the Forecast Period is selected for the Incremental Hydrologic Inflows on Subbasin category. But, there is always a Source slot and a Target slot where the Source slot is input and used to forecasted and set the Target slot. If the Forecast Period method is selected, the computational subbasin calls this method and uses the Cumulative Hydrologic Infow (Source) to forecast and set the Forecasted Cumulative Hydrologic Inflow (Target). If it is not selected, the reservoir calls this method at the beginning of the timestep and uses the Deterministic Incremental Hydrologic Inflow (Source) slot to forecast and set the Hydrologic Inflow Forecast slot (Target). The following uses the Source/Target terminology to describe the methods.

The Source slot values are input for each timestep. At the beginning of each controller timestep, the Exponential Recession method is executed. For each forecast timestep within the period of perfect knowledge, the Target is set to the Source value. For each forecast timestep after the period of perfect knowledge, the Target slot is set as described below:

$$ForecastedFlow = MAX\left[MinimumForecastedFlow, \left(Source \bullet e^{\frac{(-C)t}{T}}\right)\right]$$

where Source is the value in the Source slot at the end of the period of perfect knowledge, C is the recession coefficient, t is the elapsed time of the forecast period, and T is the total time from the end of the period of perfect knowledge to the end of the forecast period.

If the Source at the end of the period of perfect knowledge is negative, the Target at that timestep is exactly equal to the Source. However, the Source used in the recession equation, is the last positive value for Source. In the event that there is not a positive value for the Source, RiverWare issues a warning, and all values for Hydrologic Inflow Forecast within the forecast period will be set to the Minimum Forecasted Flow.

A value for the Source slot must be known at every timestep during the run. If the Target values are desired past the end of the run, there must also be values in the Source slot at timesteps past the end of the run. If values for Source are not entered past the end of the run, the Target values for these timesteps are assumed to be zero.

24.1.10.4 Coefficient and Exponent

On each timestep in the forecast period, this method will adjust the inflow hydrographs. If the Forecast Period method is selected for the Incremental Hydrologic Inflows on Subbasin category, the Cumulative Hydrologic Inflow is used to forecast and set the Forecasted Cumulative Hydrologic Inflow. If the Forecast Period methods is not selected, the input Deterministic Incremental Hydrologic Inflow slot is used to forecast and set the Hydrologic Inflow series slot.

Slots Specific to this Method

☞ Hydrologic Inflow

Type: Series
Units: FLOW
Description: flow into the reservoir that is not gaged and/or does not enter through the main channel.
Information:
I/O: Output only
Links: Not linkable

☞ Hydrologic Inflow Adjust

Type: Series
Units: FLOW
Description: optional adjustment that can be made to the calculated Hydrologic Inflow
Information:
I/O: Optional; the Hydrologic Inflow Adjust may either be input by the user or it is set to zero if it is not input.
Links: Not linkable

☞ Hydrologic Inflow Net

Type: Series
Units: FLOW
Description: Sum of Hydrologic Inflow and Hydrologic Inflow Adjust
Information:
I/O: Output only
Links: Not linkable

☞ Forecast Inflow Parameters

Type: Table
Units: NONE
Description: Table slot that contains four parameters used in the forecast inflow method. The first row contains the values for the increasing hydrograph, the second row contains values for the decreasing hydrograph. The first column contains coefficients, the second column contains exponents.
Information: 2X2 table

I/O: Input only

Links: not linkable

☞ Deterministic Incremental Hydrologic Inflow

Type: Series

Units: FLOW

Description: This slots holds a timeseries of the actual Hydrologic Inflows to the reach. These values are then adjusted by the forecast method and set on the Hydrologic Inflow slot.

Information: At the end of the run, the Hydrologic Inflow Forecast slot will be identical to this slot. If the Compute Full Run Incremental Inflows method is selected, values from the Incremental Hydrologic Inflow slot will be copied into this slot prior to the forecast. This slot is not used (i.e. inputs are ignored) if the Forecast Period method is selected. The logic below uses the Lower Bound on the Deterministic Incremental Hydrologic Inflow slot as a minimum value. This is specified slot configuration (View->Configure menu). Consider setting this value as needed.

I/O: Input or set to the values in the Incremental Hydrologic Inflow slot

Links: not linkable

☞ Forecast Period

Type: Table

Units: None

Description: Number of timesteps, not including the current timestep, that the inflow hydrograph will be adjusted.

Information:

I/O: Input only

Links: not linkable

This method uses different slots and sets different slots if the Forecast Period method is selected for the Incremental Hydrologic Inflows on Subbasin category. But, there is always a Source slot and a Target slot where the Source slot is input and used to forecasted and set the Target slot. If Forecast Period is NOT selected, the reservoir calls this method at the beginning of the timestep and uses the Deterministic Incremental Hydrologic Inflow (Source) slot to forecast and set the Hydrologic Inflow slot (Target). If it is selected, the computational subbasin calls this method and uses the Cumulative Hydrologic Inflow (Source) to forecast and set the Forecasted Cumulative Hydrologic Inflow (Target). The subbasin then computes the incremental flow and sets the value on the Hydrologic Inflow slot. The following description uses the Source/Target terminology to describe the method.

The method works as follows: on the current timestep, the Target is set equal to the Source. The method then loops through the remaining timesteps in the forecast period and sets the Target using the following formula starting at i = 1:

$$HI_i = HI_{i-1} + HI_{i-1}\frac{(KI_i - KI_{i-1})}{KI_{i-1}}((C^i)^E)$$

where KI_i is the Source at timestep i, HI_i is the Target at timestep i. The counter i represents the timestep beyond the current timestep. For example, i = 1 is the next timestep, i = 2 is the current timestep + 2 timesteps, etc. The coefficient, C, and exponent, E, are the values in the Forecast Inflow Parameters slot. If(Source(i-1) <= Source(i)), E and C are the increasing (rising) values. Otherwise, E and C are the decreasing (falling) values.

In the above formula, there is a mathematical problem if KI_{i-1} is zero. In this situation, the Target at that index is set to the known inflow at that index. This allows the simulation to continue with reasonable values for the Target. The logic uses the Lower Bound on the Target slot as the minimum value. This is specified on the slot configuration (View->Configure menu). If this value is specified it is checked, otherwise only 0.0 is used in the check. The logic is:

If (Source(i-1) = 0.0 OR ABS(Source(i-1)) < ABS(Lower Bound)))

Target(i) = Source(i)

At the end of the method, the Hydrologic Inflow Net is set equal to the Hydrologic Inflow plus the Hydrologic Inflow Adjust for each timestep in the loop. This allows the rules to be able to use the Hydrologic Inflow Net before the object dispatches (such as in the GetMaxOutflowGivenInflow() function).

A value for the Source slot must be known at every timestep during the run. If forecasted Hydrologic Inflow values are desired past the end of the run, there must also be values in the Source slot at timesteps past the end of the run. If values for Source are not entered past the end of the run, the Hydrologic Inflows for these timesteps are assumed to be zero.

24.1.11 Incremental Hydrologic Inflows on Subbasin

The Incremental Hydrologic Inflows on Subbasin category contains methods used to specify that the reservoir has cumulative inflows that must be disaggregated into incremental inflows. There are two methods: Full Run and Forecast Period. The disaggregation is actually executed from the computation subbasin containing the reservoir. For more information, click HERE (Section 7.1.22).

When the Input Hydrologic Inflow method is selected in the Hydrologic Inflow category, the Full Run becomes available. The method contains two slots: Cumulative Hydrologic Inflow and Incremental Hydrologic Inflow for the reservoir. Data must be input into the Cumulative Hydrologic Inflow slots. The computational subbasin will execute the Compute Full Run Incremental Hydrologic Inflows method and set the Incremental Hydrologic Inflow slots as input. Setting the Incremental Hydrologic Inflow slots as input prevents the slot values from being cleared in future model runs when the subbasin is disabled. The computational subbasin will first check that Full Run method has been selected on all reservoirs and reservoirs in the basin. During the model run, when the reservoir dispatches, the Full Run method will copy the value in the Incremental Hydrologic Inflow slot and set the Hydrologic Inflow slot.

When a forecast method is selected in the Generate Forecast Hydrology methods on the reservoir, a new method will be available within the Incremental Hydrologic Inflows on Subbasin category: Forecast Period. This method will contain the Cumulative Hydrologic Inflow slot. When this method is selected, the computational subbasin will execute the Compute Forecast Period Incremental Hydrologic Inflows method and set the Hydrologic Inflow slots.

24.1.11.1 None

This method is the default for the Incremental Hydrologic Inflows on Subbasin category and should be selected when hydrologic inflow data is not cumulative or the computation of incremental hydrologic inflows is not desired. There are no slots specifically associated with this method.

24.1.11.2 Full Run

The Full Run method is available from the Incremental Hydrologic Inflows on Subbasin category on the reservoirs (storage reservoir, level power reservoir, and sloped power reservoir,). This method is only available if the Input Hydrologic Inflow method is selected in the Hydrologic Inflow category. This method holds the slots necessary for the computation of incremental hydrologic inflows that is performed by the computational subbasin. The method contains two slots: Cumulative Hydrologic Inflow and Incremental Hydrologic Inflow. These slots will be accessed by the computational subbasin when executing the Compute Full Run Incremental Local Inflows method. The computational subbasin uses input Cumulative Hydrologic Inflow values to calculate and set the

Incremental Hydrologic Inflow slot. Refer to the computational subbasin's Calculate Incremental Local Inflows documentation for details, click HERE (Section 7.1.22.2). The Full Run method on the reservoir copies the results of the calculation in the Incremental Hydrologic Inflow slot values over to the Hydrologic Inflow series slot.

SLOTS SPECIFIC TO THIS METHOD

☞ **CUMULATIVE HYDROLOGIC INFLOW**

Type: Series
Units: FLOW
Description: The cumulative hydrologic inflow to the reservoir
Information: Hydrologic inflow is cumulative either (1) between headwater control points and the first reservoir in the river system and also throughout the system between two reservoirs, or (2) throughout the entire river system.
I/O: Required Input
Links: Not linkable

☞ **INCREMENTAL HYDROLOGIC INFLOW**

Type: Series
Units: FLOW
Description: The incremental hydrologic inflow to the reservoir
Information: This slot is set by the Calculate Incremental Flows method on the computational subbasin and represents the actual hydrologic inflow to the reservoir. If a Generate Forecast Hydrology method is selected, the Deterministic Incremental Hydrologic Inflow slot will be set to the values in this slot.
I/O: Computed and set with the Input flag
Links: Usually not linked

24.1.11.3 Forecast Period

The Forecast Period method is available from the Incremental Hydrologic Inflows on Subbasin category on the reservoir. This method is only available if the Forecast Hydrologic Inflows method is selected in the Hydrologic Inflow category and one of the forecasting methods (i.e. Geometric Recession, Exponential Recession, or Coefficient and Exponent) is selected in the Generate Forecast Hydrology category. If this method is selected, but the reservoir is not part of a subbasin with the appropriate methods selected, an error will be issued. The method contains two slots: Cumulative Hydrologic Inflow and Forecasted Cumulative Hydrologic Inflow. These slots will be accessed by the computational subbasin when executing the Compute Forecast Period Incremental Hydrologic Inflows method. The computational subbasin will use the user input Cumulative Hydrologic Inflow value to forecast and set the Forecasted Cumulative Hydrologic Inflow slot. It then uses this temporary value in its calculation of the incremental flows. The final result of this method (forecasted incremental hydrologic inflows) is set on the slot Hydrologic Inflow Forecast for each timestep in the forecast period. (Note: Hydrologic Inflow is set instead if the Coefficient and Exponent method is selected). Refer to the computational subbasin's Incremental Local Inflows documentation for details, click HERE (Section 7.1.22.3).

Slots Specific to this Method

☞ Cumulative Hydrologic Inflow

Type: Series
Units: Flow
Description: The cumulative hydrologic inflow to the reservoir
Information: Hydrologic inflow is cumulative either (1) between headwater control points and the first reservoir in the river system and also throughout the system between two reservoirs, or (2) throughout the entire river system.
I/O: Required Input
Links: Not linkable

☞ Forecasted Cumulative Hydrologic Inflow

Type: Series
Units: FLOW
Description: the forecasted cumulative Hydrologic inflow to the control point
Information: This slot is set by the selected Generate Forecast Hydrologic Inflows method on the control point as called from the computational subbasin. It represents the cumulative Hydrologic inflow to the control point forecasted throughout the forecast period. This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output only
Links: Usually not linked

24.1.12 Evaporation and Precipitation

The Evaporation and Precipitation category methods are used to calculate the volume of Evaporation from and Precipitation to the surface of a reservoir over the timestep. Precipitation and Evaporation are used in the mass balance equations which are solved in the dispatch methods.

Some of the methods in this category only calculate evaporation.

24.1.12.1 None

The None method is the default method for the Evaporation and Precipitation category. It should be chosen if the user does not want to include Evaporation in the mass balance equation of the Reservoir. There are no slots specifically associated with this method. No calculations are performed by this method.

24.1.12.2 Daily Evaporation

The Daily Evaporation method is used to calculate the daily evaporation volume and the flow rate of the precipitation. The daily evaporation volume is a function of the Evaporation Rate, average Surface Area, and Pan Coefficient.

SLOTS SPECIFIC TO THIS METHOD

☞ **ELEVATION AREA TABLE**

Type: Table Slot
Units: LENGTH VS. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☞ **EVAPORATION**

Type: SeriesSlot
Units: VOLUME
Description: volume of water lost to evaporation during one timestep
Information:
I/O: Output only
Links: Not linkable

☞ **EVAPORATION TABLE**

Type: Table Slot
Units: NOUNITS VS. LENGTH (PER DAY)

Description: day of the year vs. Evaporation Rate
Information: The first of January is 0.
I/O: Required Input
Links: Not linkable

☞ Pan Evaporation Coefficient

Type: Table Slot
Units: NO UNITS
Description: a fractional value between 0 and 1 that represents the portion of potential evaporation which actually occurs
Information:
I/O: Required Input
Links: Not linkable

☞ Precipitation Rate

Type: SeriesSlot
Units: LENGTH vs. TIME
Description: precipitation intensity for the given timestep
Information:
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

☞ Precipitation Volume

Type: SeriesSlot
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☞ Surface Area

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area calculated from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

The Surface Area is determined using the Pool Elevation and the Elevation Area Table. The Evaporation Rate is looked up in the Evaporation Table according to the current day of the year. Evaporation is calculated using the following equation:

$$\text{Evaporation} = \text{Evaporation Rate} \times \text{Pan Evaporation Coefficient} \times (\text{Surface Area} + \text{Surface Area}(-1))/2$$

The volume of Precipitation that occurred over the timestep is then calculated with the following equation:

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area}(-1))/2$$

where in the above equations:

Evaporation Rate = the Evaporation Rate corresponding to the current day of the year

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

24.1.12.3 Input Evaporation

The Input Evaporation method should be used when the user wants to input the Evaporation Rate directly. This Evaporation Rate is used to compute the volume of water that evaporated over the timestep. Also, the user can input the evaporation volume directly on the Evaporation slot. In that case, the Evaporation Rate is not used to calculated Evaporation.

Slots Specific to this Method

☞ Elevation Area Table

Type: Table Slot
Units: LENGTH VS. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☞ Evaporation

Type: SeriesSlot
Units: VOLUME
Description: volume of water lost due to evaporation during the timestep
Information:
I/O: Output; optional input overrides calculation
Links: Not linkable

☞ Evaporation Rate

Type: SeriesSlot
Units: LENGTH PER TIME
Description: rate at which water evaporates from the surface
Information:
I/O: Optional input, disaggregated by method as described in the Evap and Precip Rate Specification category, or defaults to 0.0 if not specified by the user.
Links: Not linkable

☞ Precipitation Rate

Type: SeriesSlot
Units: LENGTH PER TIME
Description: precipitation intensity for a given timestep
Information:
I/O: Optional input, disaggregated by method as described in the Evap and Precip Rate Specification category, or defaults to 0.0 if not specified by the user.
Links: Not linkable

☞ Precipitation Volume

Type: SeriesSlot
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☞ Surface Area

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

At the beginning of the run, the chosen method in the **Evap and Precip Rate Specification** category is executed. This category allows you to specify the rates as monthly or periodic slots.

If the user specifies Evaporation directly (via input or rules), the value will be used instead of calculating a value below.

If Evaporation is not specified, the following equation is used to compute the volume of water that evaporated from the Reservoir over the timestep:

$$\text{Evaporation} = \text{Evaporation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)}) / 2$$

The precipitation flow rate over the timestep is calculated as shown in the following equation:

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)}) / 2$$

where in the above equations:

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

24.1.12.4 Monthly Evaporation

In the Monthly Evaporation method, evaporation is calculated linearly from the Evaporation Coefficients entered for each month. This method will not work with a timestep longer than monthly. The total evaporated volume is a function of the average Reservoir Surface Area over the timestep, the Evaporation Coefficient, and the length of the timestep. The following slots are specifically associated with this method.

Slots Specific to this Method

☞ **Elevation Area Table**

Type: TableSlot
Units: LENGTH vs. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☞ **Evaporation**

Type: SeriesSlot
Units: VOLUME
Description: volume of water lost to evaporation during one timestep
Information:
I/O: Output only
Links: Not linkable

☞ Evaporation Coefficients

Type: TableSlot
Units: LENGTH PER TIME
Description: rate of evaporation for each month
Information: This slot contains one column of values. The Evaporation Coefficient for each month of the year must be input by the user beginning with the Evaporation Coefficient for January.
I/O: Required input
Links: Not linkable

☞ Precipitation Rate

Type: SeriesSlot
Units: LENGTH PER TIME
Description: precipitation intensity for the given timestep
Information: Value must be input by the user for each timestep.
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

☞ Precipitation Volume

Type: SeriesSlot
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☞ Surface Area

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area calculated from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

The Surface Area of the Reservoir is calculated based on the Elevation Area Table. The Evaporation is then calculated using the following formula:

$$\text{Evaporation} = \text{Evaporation Coefficient} \times (\text{Surface Area} + \text{Surface Area }(-1))/2 \times TimestepLength$$

The volume of Precipitation that occurred over the timestep is then calculated using the following equation:

$$\text{Precipitation} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)})/2 \times TimestepLength$$

where in the above equations:

Evaporation Coefficient = the Evaporation Coefficient for the current month

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

24.1.12.5 Pan and Ice Evaporation

The Pan and Ice Evaporation method is used to calculate the volume of evaporation with one of two methods based on the value of the Pan Ice Switch slot for each timestep. The Pan Ice Switch slot is used as an indicator of whether ice is present on the surface of the reservoir. A value of 1.0 in the Pan Ice Switch slot indicates that there is ice cover on the Reservoir that must be taken into account when Evaporation is calculated. A value of 0.0 or any number other than 1.0 in the Pan Ice Switch slot indicates that there is no ice on the surface of Reservoir. The following slots are those specifically associated with this method.

Slots Specific to this Method

☞ **Evaporation**

Type: Series
Units: VOLUME
Description: volume of water lost to evaporation during the current timestep
Information:
I/O: Output only
Links: Not linkable

☞ **Elevation Area Table**

Type: Table
Units: LENGTH vs. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☞ **K Factor**

Type: Series Slot with Periodic Input
Units: VELOCITY per TEMPERATURE_F
Description: factor relating average temperature, in degrees Fahrenheit, to evaporation rate

Information: This slot is a series slot, but the data can be input as a periodic relationship.
I/O: Optional but is required Input if the Pan Ice Switch slot is 1.0
Links: Not linkable

☞ Max Air Temperature
Type: Series
Units: TEMPERATURE IN FARENHEIT
Description: maximum air temperature during the timestep
Information:
I/O: Optional; required only if the Pan Ice Switch slot is 1.0
Links: Not linkable

☞ Min Air Temperature
Type: Series
Units: TEMPERATURE IN FARENHEIT
Description: minimum air temperature during the timestep
Information:
I/O: Optional; required if the Pan Ice Switch slot is 1.0
Links: Not linkable

☞ Pan Evaporation
Type: Series
Units: LENGTH PER TIME
Description: evaporation rate from the surface
Information:
I/O: Optional; only required if the Pan Ice Switch is 0.0
Links: Not linkable

☞ Pan Evaporation Coefficient
Type: Table
Units: DECIMAL
Description: weighing factor for pan evaporation rate
Information:
I/O: Optional; required if the Pan Ice Switch slot is 0.0
Links: Not linkable

☞ Pan Ice Switch
Type: Series
Units: No Units

Description: indicator of surface ice coverage for each timestep; **1.0** = ice; any other number or **0.0** = no ice.
Information: This slot is a series slot, but the data can be input as a periodic relationship.
I/O: Required input
Links: Not linkable

☞ PRECIPITATION RATE
Type: Series
Units: LENGTH PER TIME
Description: precipitation intensity for a given timestep
Information:
I/O: Optional; defaults to 0.0 if not specified by the user.
Links: Not linkable

☞ PRECIPITATION VOLUME
Type: Series
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

☞ SURFACE AREA
Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

☞ SURFACE ICE COVERAGE
Type: SeriesSlot
Units: DECIMAL
Description: fraction of the Surface Area which is covered by ice
Information:
I/O: Optional; only used if the Pan Ice Switch slot is 1.0. Defaults to 0.0 for any timestep not specified by the user.
Links: Not linkable

If the Pan Ice Switch slot is equal to 1.0, ice is present and the following calculation is performed to compute evaporation:

$$\text{Evaporation} = \frac{\text{Max Air Temperature} + \text{Min Air Temperature}}{2} \times \text{K Factor} \times (1 - \text{Surface Ice Coverage}) \times \text{average Surface Area} \times \mathit{Timestep}$$

If the calculated Evaporation is less than zero, the Evaporation is set equal to zero.

The Precipitation is calculated with the following equation if the Pan Ice Switch slot is equal to 1.0:

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times (1 - \text{Surface Ice Coverage}) \times \text{average Surface Area}$$

The volume of precipitation that accumulated over the timestep at the Reservoir (Precipitation Volume) is the product of the precipitation flow rate and the timestep.

If the Pan Ice Switch slot is 0.0 or any number other than 1.0, there is no ice and the following calculation is performed to compute Evaporation:

$$\text{Evaporation} = \text{Pan Evaporation} \times \text{Pan Evaporation Coefficient} \times \text{average Surface Area} \times \text{Timestep}$$

$$\text{precipitation flow rate} = \text{Precipitation Rate} \times \text{average Surface Area}$$

The volume of precipitation that accumulated over the timestep at the Reservoir (Precipitation Volume) is the product of the precipitation flow rate and the timestep.

24.1.12.6 Pan and Ice Evaporation, Current Surface Area

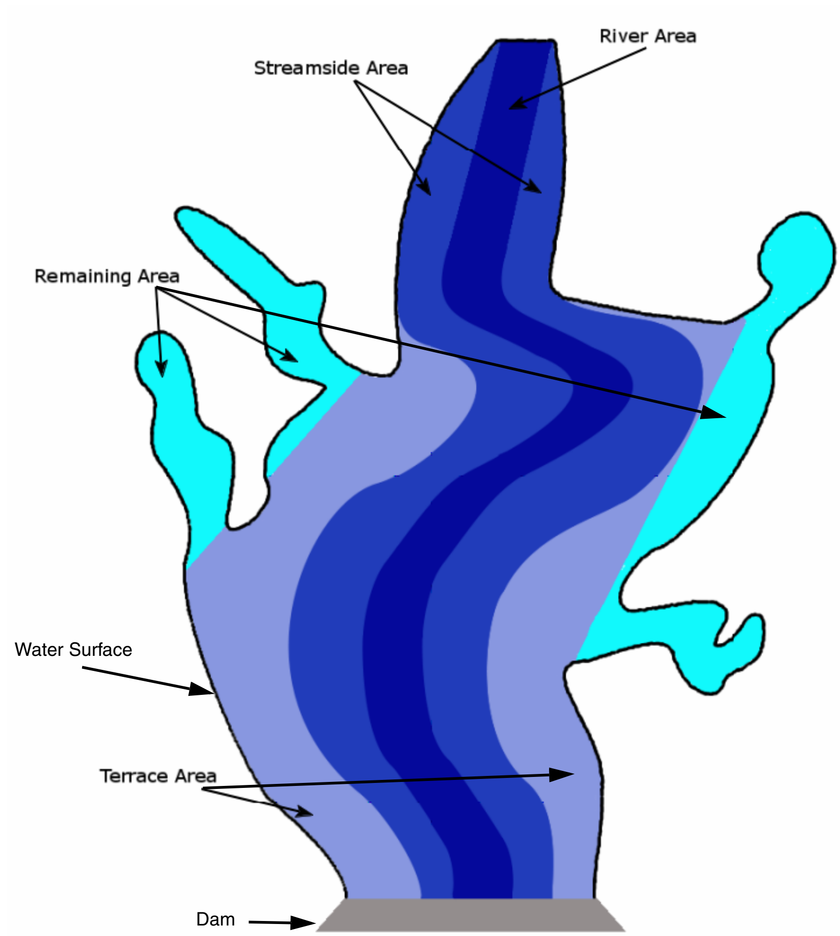
This method is exactly the same as the Pan and Ice Evaporation method. It uses the same slots, has the same required inputs and performs the same calculations. The only difference is that this method uses the instantaneous, end of timestep surface area instead of the average surface area over the timestep.

24.1.12.7 Periodic Net Evaporation

Periodic Net Evaporation computes the gross evaporation from the reservoir and then subtracts out components of evaporation that would have occurred if the reservoir had not been built. This is the net evaporation and is set in the **Evaporation** slot. Each area of the submerged reservoir is separate including:

- River
- Streamside
- Terrace, and any
- Remaining areas

Each can have a separate evaporation coefficient and possibly additional components in its computation like temperature. The area of each region is specified in a separate table relating reservoir pool elevation to each region's area. It is assumed that any precipitation that falls on the Remaining Area would have completely evaporated. The figure shows a sample of the different reservoir areas used in this method.



Slots Specific to this Method

Average Precipitation

Type: Periodic
Units: Velocity (Length per Time)
Description: Slot describing the average precipitation
Information: Typically this would have a yearly period and monthly precipitation values. It is used in the computation of Remaining Evaporation.
I/O: Required Input
Links: Not Linkable

Average Air Temperature

Type: Periodic
Units: TemperatureInFahren
Description: Slot describing the average air temperature

Information: Typically this would have a yearly period and monthly temperature values. It is used in computation of Streamside Evaporation and Terrace Evaporation.
I/O: Required Input
Links: Not Linkable

☞ ELEVATION AREA TABLE
Type: Table
Units: LENGTH VS AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required Input
Links: Not Linkable

☞ ELEVATION RIVER AREA
Type: Table
Units: LENGTH VS AREA
Description: Table relating reservoir Pool Elevation to submerged river area.
Information:
I/O: Required Input
Links: Not Linkable

☞ ELEVATION STREAMSIDE AREA
Type: Table
Units: LENGTH VS AREA
Description: Table relating reservoir Pool Elevation to submerged streamside area.
Information:
I/O: Required Input
Links: Not Linkable

☞ ELEVATION TERRACE AREA
Type: Table
Units: LENGTH VS AREA
Description: Table relating Pool Elevation to submerged terrace area
Information:
I/O: Required Input
Links: Not Linkable

☞ EVAPORATION
Type: Series
Units: VOLUME

Description: Water lost from the reservoir to evaporation. This is the net evaporation and is the value that is included in the reservoir mass balance.
Information: This is the calculated as Gross Evaporation minus Salvage Evaporation
I/O: Output only
Links: Not Linkable

☞ GROSS EVAPORATION
Type: Series
Units: VOLUME
Description: The total evaporation off the reservoir surface. This is the evaporation that is actually occurring from the reservoir.
Information: This is calculated as GrossEvaporationCoeff times SurfaceAreaAvg converted from a flow to volume.
I/O: Output only
Links: Not Linkable

☞ GROSS EVAPORATION COEFFICIENT
Type: Periodic
Units: VELOCITY (LENGTH PER TIME)
Description: A table that describes the gross evaporation coefficient as it varies periodically. This is similar to a pan evaporation coefficient.
Information:
I/O: Required Input
Links: Not Linkable

☞ RIVER EVAPORATION COEFFICIENT
Type: Periodic
Units: VELOCITY (LENGTH PER TIME)
Description: A table that describes the river evaporation coefficient as it varies periodically. This is similar to a pan evaporation coefficient.
Information:
I/O: Required Input
Links: Not Linkable

☞ SALVAGE EVAPORATION
Type: Series
Units: VOLUME
Description: The evaporation that would have occurred if the reservoir were not in place.

$$SalvageEvaporation = RiverEvaporation + StreamsideEvaporation + TerraceEvaporation + RemainingEvaporation$$

Information:
I/O: Output Only
Links: Not Linkable

☞ STREAMSIDE COEFFICIENT

Type: Periodic
Units: VELOCITYPERTEMPERATURE_F (I.E. LENGTH PER TIME PER TEMPERATURE_F)
Description: Periodic table of coefficients
Information: Typically this represents a unit depth per month per degree Fahrenheit (e.g. inches per month per degree Fahrenheit)
I/O: Required Input
Links: Not Linkable

☞ SURFACE AREA

Type: Series
Units: AREA
Description: Reservoir surface area computed from a lookup on the Elevation Area table
Information:
I/O: Output only
Links: Not Linkable

☞ TERRACE COEFFICIENT

Type: Periodic
Units: VELOCITYPERTEMPERATURE_F (I.E. LENGTH PER TIME PER TEMPERATURE_F)
Description: Periodic table of coefficients
Information: Typically this represents a unit depth per month per degree Fahrenheit (e.g. inches per month per degree Fahrenheit)
I/O: Required Input
Links: Not Linkable

The method will be passed in a current estimate of Surface Area and Average Surface Area. The latter is an average of the current estimate and previous timestep's value. In this description, it is called SurfaceAreaAvg. Similarly, StreamsideAreaAvg, RiverAreaAvg, TerraceAreaAvg and RemainingAreaAvg are all averages of the current and previous values.

In the following steps, the FlowToVolume and VolumeToFlow notation indicates that the specified expression will be converted from a flow to a volume (or vice versa) using the

timestep length. This is necessary for the units to work correctly as evaporation [Volume units] is computed as a coefficient [Length/Time units] times an area [L^2 units]. Note that in the following steps, the slots are in bold while intermediate values are not.

The method does the following:

1. Get the value from the periodic **Gross Evaporation Coefficient** slot. If not valid, issue an error.
2. Compute **Gross Evaporation**:

$$GrossEvaporation = FlowToVolume(GrossEvaporationCoefficient \times SurfaceAreaAvg)$$

3. Get the value from the periodic **River Evaporation Coefficient** slot. If not valid, issue an error.
4. Look up the Pool Elevation at t and t-1 on the **Elevation River Area** table to get the RiverArea at t and t-1. Then

$$RiverAreaAvg = \frac{RiverArea[t] + RiverArea[t-1]}{2}$$

5. Compute River Evaporation:

$$RiverEvaporation = FlowToVolume(RiverEvaporationCoefficient \times RiverAreaAvg)$$

This simulates that the river evaporation that would have occurred without the reservoir is a function of area and coefficient.

6. Get the value from the periodic **Streamside Coefficient** slot. If not valid, issue an error.
7. Look up the Pool Elevation at t and t-1 on the **Elevation Streamside Area** table to get the StreamsideArea at t and t-1. Then:

$$StreamsideAreaAvg = \frac{StreamsideArea[t] + StreamsideArea[t-1]}{2}$$

8. Get the value from the periodic **AverageTemperature** slot. If not valid, issue an error.
9. Compute Streamside Evaporation:

$$StreamsideEvaporation = FlowToVolume(StreamsideCoefficient \times StreamsideAreaAvg \times AverageAirTemperature)$$

This simulates that the streamside evaporation that would have occurred without the reservoir is a function of area, coefficient, and average air temperature.

10. Get the value from the periodic **Terrace Coefficient** slot. If not valid, issue an error.

11. Look up the Pool Elevation at t and t-1 on the **Elevation Terrace Area** table to get the Terrace Area at t and t-1. Then:

$$TerraceAreaAvg = \frac{TerraceArea[t] + TerraceArea[t-1]}{2}$$

12. Compute Terrace Evaporation:

$$TerraceEvaporation = FlowToVolume(TerraceCoefficient \times TerraceAreaAvg \times AverageAirTemperature)$$

This simulates that the terrace evaporation that would have occurred without the reservoir is a function of area, coefficient, and average air temperature.

13. Compute the average Remaining Area as:

$$RemainingAreaAvg = SurfaceAreaAvg - RiverAreaAvg - StreamsideAreaAvg - TerraceAreaAvg$$

If RemainingAreaAvg is less than zero, an error will be issued as the table data is incorrect.

14. Get the value from the periodic **Average Precipitation**. If not valid, issue an error.

15. Compute RemainingEvaporation:

$$RemainingEvaporation = FlowToVolume(RemainingArea \times AveragePrecipitation)$$

This simulates that all of the precipitation on the Remaining Area would have evaporated.

16. Compute **Salvage Evaporation**:

$$SalvageEvaporation = RiverEvaporation + StreamsideEvaporation + TerraceEvaporation + RemainingEvaporation$$

17. Compute **Evaporation** as follows:

$$Evaporation = GrossEvaporation - SalvageEvaporation$$

The **Evaporation** is then a volume that is removed from the reservoir mass balance in the dispatch method.

24.1.12.8 Single Evaporation

In the Single Evaporation method, evaporation is calculated linearly from the Single Evaporation Coefficient entered by the user. The total evaporated volume is a function of the average Reservoir Surface Area over the timestep, the Single Evaporation Coefficient, and the length of the timestep. The following slots are specifically associated with this method.

Slots Specific to this Method

☞ **Elevation Area Table**

Type: Table Slot
Units: LENGTH vs. AREA
Description: Pool Elevation vs. Surface Area
Information:
I/O: Required input
Links: Not linkable

☞ **Evaporation**

Type: SeriesSlot
Units: VOLUME
Description: volume of water lost to evaporation during one timestep
Information:
I/O: Output only
Links: Not linkable

☞ **Single Evap Coeff**

Type: Table Slot
Units: LENGTH PER TIME
Description: rate of evaporation
Information: This slot contains a single value that represents the evaporation rate.
I/O: Required input
Links: Not linkable

☞ **Precipitation Rate**

Type: SeriesSlot

Units: LENGTH PER TIME
Description: precipitation intensity for the given timestep
Information: Value must be input by the user for each timestep.
I/O: Optional; defaults to 0.0 if not input.
Links: Not linkable

PRECIPITATION VOLUME

Type: SeriesSlot
Units: VOLUME
Description: precipitation flow rate multiplied by the length of the timestep
Information: Used in the mass balance to solve for storage
I/O: Output only
Links: Not linkable

SURFACE AREA

Type: SeriesSlot
Units: AREA
Description: Reservoir Surface Area calculated from the Elevation Area Table
Information:
I/O: Output only
Links: Not linkable

The Surface Area of the Reservoir is calculated based on the Elevation Area Table. The Evaporation is then calculated using the following formula:

$$\text{Evaporation} = \text{Evaporation Coefficient} \times (\text{Surface Area} + \text{Surface Area (-1)})/2 \times TimestepLength$$

The volume of Precipitation that occurred over the timestep is then calculated using the following equation:

$$\text{Precipitation} = \text{Precipitation Rate} \times (\text{Surface Area} + \text{Surface Area (-1)})/2 \times TimestepLength$$

where in the above equations:

Evaporation Coefficient = SingleEvapCoeff entered by the user

Surface Area = the current Surface Area of the Reservoir

Surface Area(-1) = the Surface Area of the Reservoir at the previous timestep

24.1.13 Evap and Precip Rate Specification

This category allows you to choose how the evaporation and precipitation rates will be specified. The category is only available when the **Input Evaporation** method in the **Evaporation and Precipitation** category is specified.

24.1.13.1 None

This is the default method; that is, the rates must be input, set by a rule, or they default to 0.0.

24.1.13.2 Monthly Rates

This method allows you to specify the evaporation and precipitation rates as a series of monthly values for the entire run.

☞ **Evaporation Rate Monthly**

Type: Series Slot
Units: Velocity
Description: The evaporation rate for each month of the run.
Information: You must set the timestep for this series slot to be monthly. Because this slot is monthly, it is most likely different than the run timestep. As a result, if you "synchronize objects", you must select the toggle in the synchronization control to "Exclude Slots with Different Timestep from Run." This will prevent changing the timestep of this slot when other slots are synchronized.
I/O: Optional input
Links: Not linkable

☞ **Precipitation Rate Monthly**

Type: Series Slot
Units: Velocity
Description: The precipitation rate for each month of the run.
Information: You must set the timestep for this series slot to be monthly. Because this slot is monthly, it is most likely different than the run timestep. As a result, if you "synchronize objects", you must select the toggle in the synchronization control to "Exclude Slots with Different Timestep from Run." This will prevent changing the timestep of this slot when other slots are synchronized.
I/O: Optional input
Links: Not linkable

Method Details

At the beginning of run, the method disaggregates the **Evaporation Rate Monthly** and **Precipitation Rate Monthly** to the **Evaporation Rate** and **Precipitation Rate** slots,

respectively. If the timestep of the run is monthly, it uses the values directly. If the timestep of the run is less than a month, it **look ups** the month that contains the given timestep and uses that value. No interpolation is performed.

If the run timestep is annual, an error is issued.

If the two slots are not monthly but have inputs, an error is issued.

If there is no value in the monthly slot for a given month, then the rate is set to 0.0.

24.1.13.3 Periodic Rates

This method allows you to specify the evaporation and precipitation rates as a periodic relationship.

☞ Evaporation Rate Periodic

Type: Periodic Slot
Units: VELOCITY
Description: The evaporation rate as a periodic relationship.
Information: Like other periodic slots, you can choose the period and whether to interpolate or lookup.
I/O: Required Input
Links: Not linkable

☞ Precipitation Rate Periodic

Type: Periodic Slot
Units: VELOCITY
Description: The precipitation rate as a periodic relationship.
Information: Like other periodic slots, you can choose the period and whether to interpolate or lookup. If you do not wish to model precipitation, you still must enter a zero in this periodic slot.
I/O: Required Input
Links: Not linkable

Method Details

At the beginning of run, the method sets the **Evaporation Rate** and **Precipitation Rate** slots by looking up (or interpolating as configured on the periodic slot) the given timestep in the **Evaporation Rate Periodic** and **Precipitation Rate Periodic** slots, respectively. If accessing the periodic slot fails due to missing values, then an error is issued and the run stops.

24.1.14 Low Flow Releases

This category is only used to add the slots necessary for low flow release calculations. These slots are generally used by a RPL function (called MeetLowFlowRequirement) to compute the low flow releases necessary to meet the low flow requirements on control point objects.

24.1.14.1 None

This method performs no calculations and adds no slots.

24.1.14.2 Enable Low Flow Releases

This method performs no calculations. It simply adds the Low Flow Release slot and Maximum Low Flow Delivery Rate slot. Use of this method for USACE-SWD is described HERE (USACE_SWD.pdf, Section 3.7).

Slots Specific To This Method

☞ **Low Flow Release**

Type: Series Slot
Units: FLOW
Description: The portion of the Outflow that is intended to meet a low flow requirement
Information: This slot is normally computed by a RPL function (MeetLowFlowRequirement) that computes the low flow releases necessary to meet the low flow requirements on control point objects.
I/O: Usually set by a rule
Links: Not linkable

☞ **Maximum Low Flow Delivery Rate**

Type: Periodic Slot
Units: FLOW
Description: The maximum low flow delivery rate for the reservoir
Information: This value is used by the RPL function (MeetLowFlowRequirement) that determines the low flow releases from each reservoir. Low flow releases will be limited to this value.
I/O: Required input
Links: Not linkable

24.1.15 Surcharge Release

Surcharge releases methods determine releases made during the forecast period. The releases are considered mandatory, due to the volume of water in the reservoir. These methods are executed only from the solveMB_givenInflowOutflow dispatch method when the Outflow slot is set with the surcharge release (S) flag. This flag can only be set by a rule.

24.1.15.1 None

The default. Nothing is done

24.1.15.2 Flat Top Surcharge

This method follows the procedure used by the computer program SUPER and the US Army Corps of Engineers Southwest District as described HERE (USACE_SWD.pdf, Section 3.3).

Slots Specific to this Method

☞ **Operating Level Table**

Type: Periodic
Units: Time vs Length at Operating Levels
Description: Table describing the seasonal variation of elevation (storage) in a reservoir at each of the user-designated operating levels.
Information: Number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep. See HERE (Section 22.1.19) for a method to modify which operating level table is used within a run.
I/O: Required Input
Links: Not Linkable

☞ **Forecast Period**

Type: Scalar
Units: No Units
Description: number of timesteps in the forecast period.
Information: number of timesteps over which Inflows are forecast and flood control releases (including surcharge releases) are calculated; includes current timestep.
I/O: Required Input

Links: Not Linkable

☞ Top of Conservation Pool

Type: Scalar
Units: No Units
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the conservation pool.
Information: Must be lower than the Top of Flood Pool. Values must be identical for all reservoirs in the computational subbasin on which flood control is performed. May be propagated from the subbasin.
I/O: Required Input
Links: Not Linkable

☞ Rating Curves

Type: Table
Units: Storage vs Flow (Induced Surcharge) vs Flow (Free-Flow)
Description: two curves representing the induced surcharge (minimum) and free-flow (maximum) that can be released based on elevation in the reservoir. The intersection of the two curves must be defined (there must be a row where the values in the middle and third column are equal) and the points of the free-flow curve beyond the end of the induced surcharge curve must be input in the table column for the induced surcharge curve.
I/O: Required input unless the method HERE (Section 24.1.16.2) is selected.
Links: Not Linkable

☞ Surcharge Release

Type: Series
Units: Flow
Description: release required by operating when the reservoir elevation is in the surcharge pool.
Information: set by surcharge release method for all timesteps in forecast period
I/O: Output
Links: Not Linkable

☞ Minimum Mandatory Release

Type: Series
Units: Flow
Description: minimum surcharge release that can be achieved within the timestep following the induced surcharge curve
Information: calculated for each timestep in the forecast period, but the slot is set only for the current timestep.

I/O: Output only

Links: Not linkable

☞ Maximum Mandatory Release

Type: Series
Units: Flow
Description: maximum surcharge release that can be achieved within the timestep following the free-flow rating curve
Information: calculated for each timestep in the forecast period, but the slot is set only for the current timestep.
I/O: Output only

Links: Not linkable

This method is based on the methodology developed by the US Army Corps of Engineers, Southwest Division. In this method, the surcharge, or mandatory, releases, as well as minimum and maximum mandatory releases, are determined for a number of timesteps called the forecast period that will prevent overtopping of a reservoir. These releases are determined for each timestep in the Run Control Dialog.

The name of this method comes from the process of selecting releases during the forecast period to minimize future releases. In other words, release a little more today to reduce, or flatten out, future releases. This approach aims to prevent excessively large inflows from creating equally large releases by extending the time period over which these inflows are released.

This method is used by the dispatch method, solveMB_givenInflowOutflow, whenever the surcharge release flag (S) is set on the Outflow slot (this flag can only be set by a rule). When inflows, which includes the sum of upstream inflows and hydrologic inflows, are known, and the rule sets the flag for the current controller timestep, the surcharge method is launched and surcharge releases and minimum and maximum mandatory releases are calculated for each timestep in the forecast period for the current controller timestep. This process progresses through each timestep on the Run Control Dialog.

The user should be aware of the distinction between the controller timestep, or timesteps set in the Run Control Dialog, and forecast timesteps. Once the surcharge method is invoked, surcharge calculations are performed for the forecast timesteps. The controller timestep does not advance, even though the Surcharge Release slot receives output from the method. Outputs in this slot are overwritten during the next controller timestep, except for the first.

The calculation of the surcharge releases for the forecast period is accomplished by using a loop structure once the surcharge method is invoked by the dispatch method mentioned above. Calculations for each timestep in the forecast period are exactly the same except for changes in the starting storage volume and inflows. The following sections provide a detailed description of the process that is used to calculate the surcharge release and minimum and maximum mandatory releases within.each loop. There a three main sections: calculations of the minimum and maximum mandatory releases, which bound the surcharge release, and the calculation of the surcharge release.

The first step is to determine the “forecast storage”. This volume is the initial storage (storage at end of previous computational timestep if this is the first time through the loop, or the storage at the end of the previous forecast timestep for all subsequent loops) plus the total inflow minus any surcharge releases for all forecast timesteps prior to the current forecast timestep. The total inflow is the sum of the upstream inflows plus hydrologic inflows for the current forecast period. For the first forecast timestep, the forecast storage is simply the storage in the reservoir on the previous controller timestep. On subsequent forecast timesteps, the forecast storage includes total inflows through the current forecast timestep and surcharge releases through the previous forecast timestep.

A table interpolation of the Elevation Volume Table is then done to determine the forecast elevation. If the forecast elevation is less than the lowest capacity value input in the induced surcharge curve, which is the middle column on the Rating Curves slot, the minimum mandatory release is zero and the program progresses to determine the maximum mandatory release. Otherwise it continues to determine the minimum mandatory release.

Calculation of the minimum mandatory release

If the forecast elevation falls between the highest and lowest capacity (or elevation) values on the induced surcharge curve, the program determines a discharge rate corresponding to the forecast elevation by interpolation of capacity values from the induced surcharge curve. This is the minimum that can be released based on the forecast elevation. If the forecast elevation is greater than the highest capacity value, the curve must be extended, i.e., there is no extrapolation of the curve. In a similar fashion, the total inflow is used in an interpolation of the induced surcharge curve to determine a corresponding elevation. If this elevation corresponding to the total inflow is less than the minimum discharge rate based on the forecast elevation, the pool is falling. For the opposite case, the pool is rising. If the total inflow and the minimum release corresponding to the forecast elevation are equal, the minimum release is set to this discharge rate and the program goes on to determine the maximum release.

Falling pool: If the pool is falling (the forecast inflow is less than minimum discharge corresponding to the forecast elevation), the program uses a specific portion of the induced surcharge curve to determine the minimum mandatory release. The set of points includes all points between and including the point on the curve corresponding to the total inflow for the current forecast timestep to the minimum discharge based on the forecast elevation. Using this portion of the induced surcharge curve, the program calculates how long it will take the reservoir to move from the forecast elevation to an elevation corresponding to the total inflow or how much volume can be discharged within one timestep, whichever is less. In the case of a falling pool, the process involves moving down the curve. For the opposite case (rising pool), the process will be reversed, moving “up” the curve since the discharge rate corresponding to the forecast elevation is less that the total inflow. In either case, the program begins at the curve point corresponding to the forecast storage and progresses to that point corresponding to the total inflow.

The process of progressing through the curve point is as follows:

First, the average of the discharge rates corresponding to the first two (or highest) curve points is found. Then the time it would take to release a certain volume at some discharge rate is determined. This volume is the difference in storage values from the curve points that correspond to the two discharges used to calculate the average discharge rate. The time is calculated by the following formula:

$$Time = \frac{Storage(i) - Storage(i-1)}{(TotalInflow - AverageDischarge)}$$

where

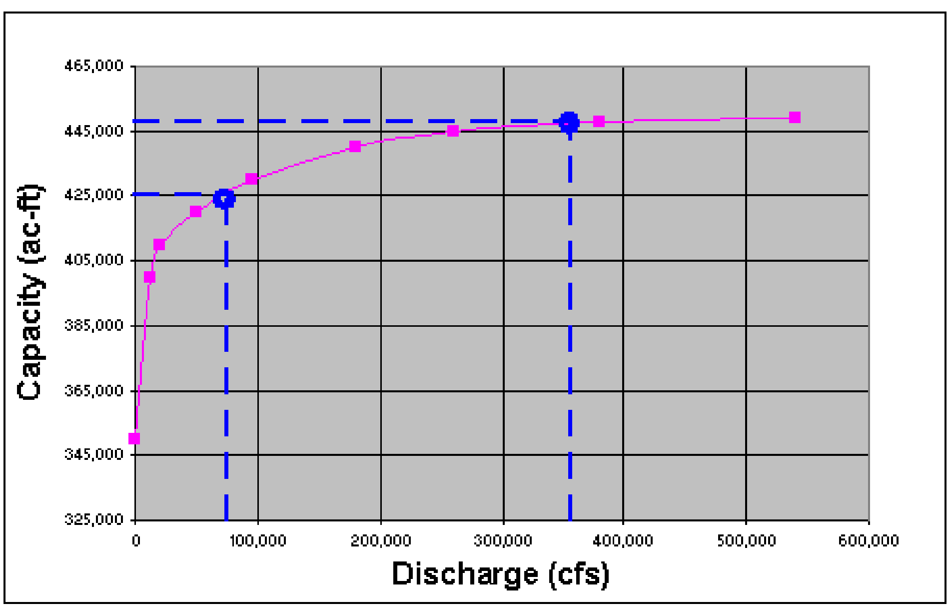
Storage(i) and Storage(i-1) are the storage values corresponding to the two discharge values (the difference is the volume that is being released). The value of Forecast Inflow - Average Discharge represents the net discharge.

If this time is less than one timestep's time, the program stores the time and the total volume that could be released in this time at the average discharge, which equal Time * Average Discharge. The program then moves to the next two curve points, gets another average discharge rate corresponding to these two curve points and again determines the time it would take to drop from one capacity value to the next at the net discharge rate. This new time plus the time for the previous two curve points is now the new total time and is again checked to see if it is greater than one timestep. If less still less than one timestep, the total time and total volume are stored (replacing the previous values) and the program moves to the next two curve points. This process continues until either the point corresponding to the total inflow is reached or the total time has reached or exceeded one timestep, whichever occurs first.

If the total time at this point is exactly equal to one timestep, the calculations are done and the minimum mandatory release equals the total volume divided by the time of one timestep. If the total time is greater than one timestep's time, the total time from the previous iteration is used to see how much time was left before reaching a total time of one timestep. Given the average discharge rate for the current iteration, the volume that could be released within the time remaining is determined. This volume is then added to the total volume and the final minimum mandatory discharge rate is calculated as described above.

If the computation gets through all the curve points, i.e., the curve point corresponding to the total inflow, with a total time that is less than or equal to one timestep, the total volume is increased by adding to the previously stored total volume to (Time of 1 timestep - total time)*total inflow. Then, the minimum mandatory release is calculated by dividing the total volume by the time of one timestep.

Example of minimum mandatory release w/ falling pool: Below is a plot of the induced surcharge curve. Assume that the timestep is a day, there is a falling pool, the total inflow for the current forecast timestep = 80,000 cfs, and the minimum release that the reservoir can make corresponding to the forecast storage of 450,000 ac-ft. = 365,000 cfs. Therefore, the portion of the curve that is used to determine the minimum mandatory release is bracketed by these two discharge points (donuts on plot), and the three actual curve points between these. First, the average discharge rate of 365,000 and 260,000 cfs = 312,500 cfs is determined. Then the volume between the two storage values corresponding to these two discharge points (450,0000 - 445,000 = 5,000 ac-ft = 217,800,000 cu. ft.) is calculated. Then, the time it takes the reservoir to release this volume of water at the calculated average discharge rate (217,800,000cu.ft./312,500 cfs = 0.26 hours) is determined. Since this time is less than 24 hours (one timestep), the total time of 0.26 hours and the total volume = 0.26*312,500 = 5,000 ac-ft is stored and the next two discharge points are used. Using these two points (260,000 and 180,000), the average discharge rate = 220,000 cfs, the volume between storage values corresponding to these two discharges = 445,000 - 440,000 = 5,000 ac-ft, and the time to release that volume at the net discharge rate = 0.43 hours. This total time = 0.69 is still less than 24 hours, so the program moves to the next two points (180,000 and 95,000) after storing the total volume, which = 10,000 ac-ft. The average discharge for the next two points is 137,500 cfs and the storage volume between these points is 440,000 - 430,000 = 10,000 ac-ft. The time to release this volume at the net discharge rate is 2.10 hours, so the total time is now 2.79 hours, which is still less than 24 hours. The total volume that is stored now = 20,000 ac-ft). The final two discharge points are 95,000 and 80,000 cfs, with an average of 87,500 cfs. The volume between these points is 5,000 ac-ft (430,000 - 425,000 ac-ft). The time to release this volume is 8.07 hours, the total time is 10.86 hours, and the total volume = 25,000 ac-ft). Although still less than 24 hours, all of the specific points of the curve have been used. The final total volume is the sum of the previous total of 25,000 ac-ft + (24hrs - 10.86)*total inflow. The minimum mandatory release is this new total volume divided by 24 hours.



Rising Pool: If the pool is rising (total inflow greater than discharge corresponding to the forecast storage), the calculation is the same as for the case of a falling pool except that the progression through the curve point will be up the curve instead of down the curve. The lowest point is the minimum discharge that must be released corresponding to the forecast storage and the highest point is the total inflow for the current forecast day.

Calculation of the maximum mandatory release

The maximum release is determined in the same manner as the minimum release except that the free-flow rating curve is used instead of the induced surcharge curve. The program again determines if the pool is be rising or falling, then progressed through the specific curve points to calculate the maximum mandatory release.

Calculate surcharge releases

Once the maximum and minimum mandatory releases have been determined, the program computes surcharge release to “flat top” the incoming flood. The objective is to have slightly higher mandatory releases sooner than later to try to reduce the surcharge releases that would need to be made later as inflows continue to come into the reservoir. The program looks ahead in the forecast period to anticipate high flows with higher releases in order to reduce the maximum release needed in the future to keep the pool at or below the induced surcharge curve until such time as the top of the induced surcharge curve is reached.

Bracket mandatory release on induced surcharge curve: To determine the mandatory release, the program steps through the points on the induced surcharge curve to determine what the surcharge release must be in order to reduce the maximum release at some time in the future when the reservoir is expected to be the fullest.

First, the available storage is calculated as the difference between the forecast storage and the storage at the first induced surcharge point. Then, the highest storage in the reservoir over the remainder of the forecast period is calculated using the remaining inflow hydrograph and a constant release equal to the discharge rate corresponding to the currently considered point on the induced surcharge curve. If the highest storage is greater than the storage available, the program continues to the next point on the induced surcharge curve and recalculates the available storage and the highest forecast storage; again using the remaining inflow hydrograph and the release associated with that point. Moving to each new curve point effectively increases the release, there by reducing the maximum storage required and increasing the available storage. This process of checking each of the points on the induced surcharge curve continues until the highest forecast storage is less than the available storage.

Before moving to another point on the curve, the program stores the available storage, the highest forecast storage, and the discharge rate to be used for future interpolation.

If the final, or highest, curve point is reached and the available storage is still less than the highest forecast storage, the program execution is terminated and a request is made for extending the curve.

Once a curve point is found that satisfies the condition that the highest forecast storage is less than the available storage, the program is considered to have found the general vicinity of the appropriate value of the surcharge release, or in other words bracketed the portion of the curve. At this time, there are two pairs of stored values for each of the following parameters: available storage, the highest forecast storage, and the discharge rate.

Use forecast inflows to zoom in on solution: Having bracketed the part of the induced surcharge curve that contains the appropriate surcharge release, it is now necessary to find the point in that range for which the available storage is just equal to the highest forecast storage. This part of the program uses inflow values from the forecasted inflow array as intermediate discharge test points. The purpose of looking through the forecasted inflows narrows the solution for the right surcharge release.

The program scans through the forecast period to see if any inflows fall between the two discharge values that have been stored. If any fall in there, the smallest is stored. Then, this inflow value is used similarly to how the discharge values from the various curve points were used above to determine the available storage and highest forecast storage if the inflow is assumed to be the release. This is followed by a check to see if the available storage is less than the highest forecast storage. If so, the program looks to see if another higher inflow within the forecast period can satisfy the condition and, if found, available storage and highest forecast storage is recalculated. However, at the next point, the lower limit within which the inflow must fit above is raised to the inflow value that was used previously. If no inflow is large enough to make the available storage larger than the required storage, continue on with the two discharge values from the curve that were stored after the solution was initially bracketed.

If there are no inflows within the forecast period that fall in the range of the two discharge points on the curve or if no inflow values during the forecast period are large enough to make the available storage large enough to exceed the highest forecast storage, the algorithms goes to the next section where an interpolation is used to find the surcharge release for the current forecast day using points off the curve.

Interpolate to find surcharge release for current forecast day: The induced surcharge curve has been bracketed by two points. At the higher point on the curve, the discharge satisfies the condition that the available storage >= highest forecast storage. At the lower point, the condition is not satisfied. (The discharge values may be specified input points from the induced surcharge curve, or may be points found as described in the previous section from the forecasted inflow array.) Once the condition of available space being larger than or equal to the required storage is satisfied, the extra available storage is calculated for the set of points that satisfied the condition (available storage - highest forecast storage) and the extra volume needed to store the highest forecast storage for the set of points that didn’t satisfy the condition (highest forecast storage - available storage). Finally, using the extra space, extra storage needed, and the two discharge points, an interpolation is done to find the surcharge release for the current forecast day. The resulting surcharge release value is that which would very closely match the available storage and highest forecast storage, given that release rate.

Compare surcharge release with maximum and minimum release values: Once the surcharge release is determined, it is compared to the maximum and minimum mandatory releases calculated earlier. If the surcharge release is greater than or less than these limits, respectively, it is set to the limit. In addition, the storage of the reservoir is reduced to reflect the surcharge release for the current forecast timestep. This storage for the current forecast timestep is used as the initial storage for the calculation of forecast storage on the next timestep in the forecast period.

Check surcharge release against conservation pool: The final section in this method involves a check to see of the surcharge release results in a pool level below top of conservation pool. The reservoir storage with the surcharge release already made through current forecast day is checked to see if it is above or below the conservation pool. If below, the surcharge release is set to that which would bring the storage down to the conservation pool. If the storage is already below the conservation pool before this release is made, the surcharge release for the current forecast timestep is zero. If above the conservation pool, the surcharge release is left as is.

The program then moves to the next forecast day. After the surcharge releases and minimum and maximum mandatory releases have been determined for every timestep in the forecast period, the calculations are complete.

Finally, the method selected in the Elevation Max Duration Constraints category HERE (Section 24.1.17) is executed. This may increase the Surcharge Release to avoid the reservoir from exceeding a desired elevation for more than the desired number of timesteps.

24.1.15.3 Induced Surcharge Curve

This method follows the procedure used by the US Army Corps of Engineers Kansas City office to determine surcharge releases during a forecast period for reservoirs with tainter gates.

SLOTS SPECIFIC TO THIS METHOD

☞ OPERATING LEVEL TABLE

Type: Periodic
Units: TIME VS LENGTH AT OPERATING LEVELS
Description: Table describing the seasonal variation of elevation (storage) in a reservoir at each of the user-designated operating levels.
Information: Number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each

column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep. See HERE (Section 22.1.19) for a method to modify which operating level table is used within a run.

I/O: Required Input

Links: Not Linkable

☞ Operating Level

Type: Series
Units: None
Description: The computed operating level
Information: This slot is computed using the pool elevation and the Operating Level Table
I/O: Output only
Links: Not Linkable

☞ ForecastPeriod

Type: Scalar
Units: No Units
Description: number of timesteps in the forecast period.
Information: number of timesteps over which Inflows are forecast and flood control releases (including surcharge releases) are calculated; includes current timestep.
I/O: Required Input
Links: Not Linkable

☞ Top of Flood Pool

Type: Scalar
Units: No Units
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the flood pool.
Information: Values must be identical for all reservoirs in the computational subbasin on which flood control is performed. May be propagated from the subbasin.
I/O: Required Input
Links: Not Linkable

☞ Top of Conservation Pool

Type: Scalar
Units: No Units
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the conservation pool. Must be lower than the Top of Flood Pool.

Values must be identical for all reservoirs in the computational subbasin on which flood control is performed. May be propagated from the subbasin.

I/O: Required Input

Links: Not Linkable

☞ Induced Surcharge Curve

Type: Table

Units: Flow (Inflow) vs Length (Elevation) vs Flow (Surcharge Release)

Description: family of curves that relate elevation and surcharge release for various inflow values.

I/O: Required Input

Links: Not Linkable

☞ Surcharge Release

Type: Series

Units: Flow

Description: release required by operating when the reservoir elevation is in the surcharge pool.

Information: set by surcharge release method for all timesteps in forecast period

I/O: Output

Links: Not Linkable

☞ Recession Target

Type: Series

Units: Length

Description: pool elevation at which surcharge operations are suspended.

Information: once pool drops to this elevation, Surcharge Release operations will terminate.

I/O: Optional, can be input, or will default to the elevation of the bottom of surcharge pool (top of flood pool).

Links: Not Linkable

☞ Gate Opening Curve

Type: Table

Units: Length (Elevation) vs Flow vs Length(Gate Opening).

Description: This is a family of curves of that relate pool elevation and discharge for various gate openings. This must be specified as a 3D table as described HERE (Three-dimensional data format).

I/O: Required Input

Links: Not Linkable

☞ **Gate Opening**

Type: Series
Units: No Units
Description: gate opening calculated for each timestep.
I/O: Output
Links: Not Linkable

☞ **Storage Tolerance**

Type: Scalar
Units: Volume
Description: allowable difference between calculated average storage and beginning of timestep storage.
Information:
I/O: Required Input
Links: Not Linkable

This method determines a surcharge release for each timestep in the forecast period. The process begins by determining a Surcharge Release value from the Induced Surcharge Curve, given Inflow (net) and current Pool Elevation (end of previous timestep for the first iteration). Given the Surcharge Release value, the Gate Opening is determined by interpolation of the Gate Opening Table. The Gate Opening is not allowed to be reduced if the reservoir is in surcharge operations (Surcharge Release > 0). If the Gate Opening from the Curve is less that the current Gate Opening, the current Gate Opening is maintained, but a new Surcharge Release is determined given that gate opening by interpolation of the Gate Opening Curve. Otherwise, the Surcharge Release from the Induced Surcharge Curve is used.

The reservoir is then mass balanced using the Inflow (net) and Surcharge Release to determine the end of period Storage. If the Surcharge Release drops the reservoir below the Recession Target, a new Surcharge Release is determined that would drop the reservoir to exactly the Recession Target and the mass balance would resolve for the end of period Storage. Then an average Storage is determined using the beginning and ending period Storages. If the difference between the current storage and the new average storage is greater than the Storage Tolerance, the process is repeated using the new average storage to compute the new current pool elevation and a new Surcharge Release. Otherwise, the method moves on to the next forecast timestep.

24.1.15.4 Pass Inflows

This method is designed to pass the inflows to the reservoir whenever the pool elevation enters the surcharge pool. If the inflows cannot be passed (due to the physical limitations of the outlet works) the surcharge release is set equal to the max outflow. Then the gates remain open until the surcharge pool is evacuated.

Slots Specific to this Method

☞ Operating Level Table

Type: Periodic
Units: Time vs Length at Operating Levels
Description: Table describing the seasonal variation of elevation (storage) in a reservoir at each of the user-designated operating levels.
Information: Number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep. See HERE (Section 22.1.19) for a method to modify which operating level table is used within a run.
I/O: Required Input
Links: Not Linkable

☞ Operating Level

Type: Series
Units: None
Description: The computed operating level
Information: This slot is computed using the pool elevation and the Operating Level Table
I/O: Output only
Links: Not Linkable

☞ ForecastPeriod

Type: Scalar
Units: No Units
Description: number of timesteps in the forecast period.
Information: number of timesteps over which inflows are forecasted and surcharge releases are calculated; includes current timestep.
I/O: Required Input
Links: Not Linkable

☞ Top of Flood Pool

Type: Scalar
Units: No Units
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the flood pool.

Information: Values must be identical for all reservoirs in the computational subbasin on which flood control is performed. May be propagated from the subbasin.
I/O: Required Input
Links: Not Linkable

☞ TOP OF CONSERVATION POOL
Type: Scalar
Units: NO UNITS
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the conservation pool. Must be lower than the Top of Flood Pool. Values must be identical for all reservoirs in the computational subbasin on which flood control is performed. May be propagated from the subbasin.
I/O: Required Input
Links: Not Linkable

☞ SURCHARGE RELEASE
Type: Series
Units: FLOW
Description: the computed surcharge release
Information: set by surcharge release method for all timesteps in forecast period
I/O: Output
Links: Not Linkable

This method is executed for every timestep in the forecast period when the surcharge release flag is set on the Outflow slot. The surcharge release is set equal to the total inflow to the reservoir (Inflow plus Hydrologic Inflow Forecast) if the pool elevation will exceed the top of the flood pool. If the inflow cannot be passed due to outlet constraints, the surcharge release is set equal to the maximum outflow. If this happens, the gates remain fully open and the surcharge release is set equal to the max outflow until the surcharge pool is evacuated. If the computed surcharge release will bring the pool elevation below the top of the flood pool, it will be reset to bring the pool elevation exactly to the top of the flood pool.

24.1.15.5 Specified Surcharge

This method is used to compute the surcharge release based on a user specified value. As always, the surcharge release is limited to the physical maximum outflow from the reservoir.

SLOTS SPECIFIC TO THIS METHOD

☞ OPERATING LEVEL TABLE
Type: Periodic
Units: TIME VS LENGTH AT OPERATING LEVELS

Description: Table describing the seasonal variation of elevation (storage) in a reservoir at each of the user-designated operating levels.

Information: Number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep. See HERE (Section 22.1.19) for a method to modify which operating level table is used within a run.

I/O: Required Input

Links: Not Linkable

☞ Operating Level

Type: Series

Units: None

Description: The computed operating level

Information: This slot is computed using the pool elevation and the Operating Level Table

I/O: Output only

Links: Not Linkable

☞ ForecastPeriod

Type: Scalar

Units: No Units

Description: number of timesteps in the forecast period.

Information: number of timesteps over which inflows are forecasted and surcharge releases are calculated; includes current timestep.

I/O: Required Input

Links: Not Linkable

☞ Top of Flood Pool

Type: Scalar

Units: No Units

Description: Operating level (as defined in Operating Level Table) corresponding to the top of the flood pool.

Information: Values must be identical for all reservoirs in the computational subbasin on which flood control is performed. May be propagated from the subbasin.

I/O: Required Input

Links: Not Linkable

☞ Top of Conservation Pool

Type: Scalar
Units: No Units
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the conservation pool.
Information: Must be lower than the Top of Flood Pool. Values must be identical for all reservoirs in the computational subbasin on which flood control is performed. May be propagated from the subbasin.
I/O: Required Input
Links: Not Linkable

☞ Surcharge Release

Type: Series
Units: Flow
Description: the computed surcharge release
Information: Set by surcharge release method for all timesteps in forecast period
I/O: Output
Links: Not Linkable

☞ Specified Surcharge release

Type: Scalar Slot
Units: Flow
Description: The user input surcharge release
Information: This value is used to set the surcharge release slot whenever the pool elevation is in the surcharge pool but less than the Critical Elevation
I/O: Required Input
Links: Not Linkable

☞ Critical Elevation

Type: Scalar Slot
Units: Length
Description: the elevation at which the surcharge release changes from the Specified Surcharge Release to the maximum release
I/O: Required Input
Links: Not Linkable

This method is executed for every timestep in the forecast period when the surcharge release flag is set on the Outflow slot. The surcharge release is set equal to the Specified Surcharge Release if the pool elevation will exceed the top of the flood pool. If the pool elevation will exceed the Critical Elevation, the surcharge release is set equal to the maximum outflow. If this happens, the gates remain fully open and the surcharge release is set equal to the max outflow until the surcharge pool is evacuated. If the computed surcharge release will bring the pool elevation below the top of the flood pool, it will be reset to bring the pool elevation exactly to the top of the flood pool.

24.1.16 Rating Curves Modification

This category is dependent on having **Flat Top Surcharge** method selected in the **Surcharge Release** category. Methods are used to optionally modify the Rating Curves table.

24.1.16.1 None

This is the default method, no modification to the Rating Curves is made.

24.1.16.2 Specify Rating Curves using Elevation

When the **Specify Rating Curves using Elevation** method is selected the following slot is added:

☞ **RATING CURVES USING ELEVATION**
Type: Table Slot
Units: ELEVATION VS FLOW VS FLOW
Description: This table represents the Rating Curves Table with Elevation as the independent variable instead of Storage as shown on the Rating Curves table. Thus, The table relates Elevation, Induces Surcharge, and Free Flow Rating Curve.
Information: Specified only
I/O: Not linkable

When this method is selected, at the beginning of the run, the reservoir will generate the Rating Curves based on the Rating Curves using Elevation slot as follows:

1. Copy all of the data in the Rating Curves using Elevation table to the Rating Curves table. This will resize and/or rebuild the table as necessary.

2. For each row in the Rating Curves using Elevation table, the Elevation will be looked up on the Elevation Volume Table to determine the Storage.

3. The Storage will then be written to the Rating Curves table. The Induced Surcharge Curve and Free Flow Rating values were already copied from the Rating Curves using Elevation table.

4. The Rating Curves table will be marked as “has Source Slot”. This makes the slot read-only with cross hatching over the value. It also adds: “Note: these values are automatically generated based on Reservoir.Rating Curves using Elevation”.

Within the Flat Top Surcharge method, the original Rating Curves (storage based) tables will be used for all computations.

24.1.17 Elevation Max Duration Constraints

This category is dependent on the Flat Top Surcharge method being selected in the Surcharge Release category:

24.1.17.1 None

This is the default method in this category. There are not maximum duration constraints on Pool Elevation.

24.1.17.2 Constant Additional Surcharge Release

The Constant Additional Surcharge Release method allows the user to specify 1 or more elevation max duration constraints. The method is executed at the end of the Flat Top Surcharge method to compute the additional volume of water to meet the maximum elevation duration constraints. This volume is then released as a constant (additional) flow of water from the current timestep to the violation date.

The new method has two method specific slots:

☞ **ELEVATION MAXIMUM DURATION TABLE**

Type: Table Slot
Units: LENGTH VS NUMBER OF TIMESTEPS VS NUMBER OF TIMESTEPS
Description: The input data in this table drive this method.
Information: It has three columns: (1) Pool Elevation (length): the elevation to which the constraint applies. Rows must be in order of decreasing elevations. (2) Maximum Duration: the maximum number of timesteps this reservoir can be above this row's Pool Elevation. (3) Reset Duration: the minimum number of timesteps the reservoir elevation must remain at or below this row's Pool Elevation in order to reset the duration counter; i.e., how long must the Pool Elevation remain low to be considered not “above this row's Pool Elevation”.
I/O: Required Input
Links: NA

☞ **ELEVATION MAXIMUM DURATION RELEASE**

Type: Aggregate Series Slot
Units: FLOW
Description: The additional flows added to Surcharge Release explicitly for the purpose of meeting the Pool Elevation maximum duration constraints.
Information: The first column is the total for all constraints, the other columns correspond to single constraints. The column name are based on the row labels in the table described above.
I/O: Output only

Links: NA

This method is executed as part of the Flat Top Surcharge. After the Surcharge Release calculation has calculated the reservoir releases necessary for evacuating the surcharge pool over the forecast period, this method determines if this proposed release schedule would violate a Pool Elevation maximum duration constraint. If so, then the surcharge release is increased as necessary to avoid violating the constraint.

The additional flow is calculated as the volume of water which would need to be released to avoid violation of the constraint, and adding that volume to the release schedule in equal increments, i.e. constant additional release. This calculation is performed for each row of the Elevation Maximum Duration Table. Each iteration may add water to the proposed release.

Continuing with the example above, assume that at timestep t the reservoir computes the Surcharge Releases and that (given these releases) the reservoir's storage will have been between 800 and 900 m for 5 days at timestep t + 2. The difference in storage between the projected Pool Elevation at t + 2 and 1000 m is 300 m3. RiverWare will then add 100 m3 to the surcharge releases (and Outflow) for t, t + 1, and t + 2 to bring the elevation down to 1000 m at t + 2.

This method sometimes produces results that are counter-intuitive. Following are some items to consider when looking at results:

- **Multiple constraints lead to non constant release patterns:** This method determines the constant additional release required to meet each constraint. The actual additional flow that will be added to the surcharge release is the sum of the values for each constraint. Therefore, the final release schedule may not be a constant value if there are multiple constraints in effect.

- **Surcharge pattern leads to non constant release patterns:** The originally computed Surcharge Release schedule is not constant, so adding flows to the surcharge release will result in a non-constant release schedule.

- **Forecast values are overwritten at each timestep:** The Surcharge Release method, including this method, is executed at each timestep and will update the values in the Surcharge Release and other slots throughout the forecast period. Therefore, at the end of the run, the results may not be obvious what is happening. If you really want to see the schedules used for a particular timestep, pause the run after that timestep has executed, then look at slot values or the special results in the model Run Analysis tool, **HERE (USACE_SWD.pdf, Section 5.5)**.

- **Constraining elevations may not be exactly met:** Surcharge Release operations do not include Evaporation or Precipitation in the computation. Similarly, reservoir diversions (withdrawals) are performed after surcharge operations. Once the other operations execute and the reservoir dispatches and solves, the resulting elevation may be slightly different than the values used in the Surcharge and elevation max duration computations.

This will manifest in not exactly reaching the constraining elevations. Because diversions (and evaporation) remove water from the reservoir, typically the elevations will be slightly below the constraining elevations.

- **Duration constraints still aren't met:** Maximum outflow constraints are applied after this additional release schedule is computed. Thus, even if the an additional amount is required, it may not be released and the target elevation may not be achieved. The proposed values are stored in the Surcharge Release slot, the actual maximums would be stored in the Outflow slot.
- **Elevations are unexpectedly at (or below) the constraint:** The algorithm looks at the pool elevation after forecasting and surcharge have executed. (This pre-constrained pool elevation is currently not available but looking at the forecasted inflows gives some indication of the values.) It then determines the additional flow to prevent violating the constraint at any time in the forecast period. If the elevation is already at the constraint, then the additional flow will likely keep the elevations at or even below the constraint. The method will add releases to the **current timesteps** (and future) timestep. There is no mechanism to delay releasing water to wait until the flood actually arrives. Thus, if it seems like the elevation should stay above the constraint, the algorithm will start releasing to make room for forecasted inflows.

  For example, when a large flood is within the period of perfect knowledge, then the algorithm may then need to release a large volume to meet a constraint at some point in the forecast period. This may cause the reservoir to go below the constraint to make room for the incoming flood (i.e. dig a hole for the incoming flood). But, the elevation should end up at the constraining elevation on the constraining timestep.
- **There are still violations in the forecast period:** Only one violation per constraint is considered, though in theory there could be another violation later in the forecast period.

24.1.18 Flood Control Release

Flood control releases on reservoirs cause dependent slots to be available for use by the predefined flood control rule function.

24.1.18.1 None

The default. No dependent slots.

24.1.18.2 Operating Level Balancing

Choose this method when you wish to use rules to perform Operating Level Balancing flood control on a computational subbasin (**HERE (Section 7.1.3.3)**) of which this reservoir is a member. Use of this method for USACE-SWD is described **HERE (USACE_SWD.pdf, Section 3.6)**.

Slots Specific to this Method

☞ **Forecast Period**

Type: Scalar
Units: No Units
Description: Number of timesteps in the forecast period.
Information: Number of timesteps over which storages are forecasted and surcharge releases are calculated; includes current timestep. Values must be identical for all reservoirs in the computational subbasin on which flood control is performed. May be propagated from the subbasin.
I/O: Required Input
Links: Not Linkable

☞ **Balance Period**

Type: Scalar
Units: No Units
Description: Number of timesteps in the balance period.
Information: Number of timesteps into the future at which the flood control inspects the storage of the reservoir to determine the goal for flood control releases. The goal is to release all the water in the flood pool at this future timestep (the current timestep is 1). Values must be identical for all reservoirs in the computational subbasin on which flood control is performed. May be propagated from the subbasin.
I/O: Required Input
Links: Not Linkable

☞ Target Balance Level

Type: Series
Units: No Units
Description: Balance level assigned to this subject reservoir by a controlling key control point, during execution of the control point's Operating Level Balancing method in the Key Control Point Balancing category.
Information: This slot is written as a result of the FloodControl predefined function HERE (RPLPredefinedFunctions.pdf, Section 40).
I/O: Output only - set by a rule.
Links: Not Linkable

☞ Max Flood Control Release

Type: Series
Units: Flow
Description: Upper bound on flood control release, used by Operating Level Balancing flood control algorithm, and computed by Operating Level Balancing key contorl point balancing method on key control points. The flood control algorithm applies this upper bound on intermediate passes of the algorithm, but not on the last pass.
Information: This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Rule output.
Links: Not linkable.

☞ Flood Control Release

Type: Series
Units: Flow
Description: The release made from the reservoir as a result of the application of a flood control rule.
Information: The value for this slot is computed by the flood control predefined rules function. The slot's value is meant to be assigned by the rule.
I/O: Rule output.
Links: Not Linkable

☞ Flood Control Minimum Release

Type: Series Slot
Units: Flow
Description: This slot holds the user specified (via a rule) minimum flood control release. This slot is used by the Operating Level Balancing algorithm as a minimum release, but is not included in the Flood Control Release slot's value. Click

HERE (USACE_SWD.pdf, Section 3.4) for more information on the intended use of this slot.

Information: This value should be set (along with Outflow) prior to calling the flood control method so that water is allowed to route downstream.

I/O: Typically set by a rule

Links: Not Linkable

☞ Proposed Flood Control Release

Type: Series

Units: FLOW

Description: The running value of the flood control release calculated by the Flood Control Release method on the Computation Subbasin.

Information: This slot is used to hold “proposed” flood control release during the flood control calculations. This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool **HERE (USACE_SWD.pdf, Section 5.5)**.

I/O: Output

Links: Not linkable

☞ Surcharge Release

Type: Series

Units: Flow

Description: The release made from the reservoir as a result of the application of surcharge release rule.

Information: The values in this slot are used by the Operating Level Balancing flood control method invoked by the predefined rules function.

I/O: Rule output.

Links: Not Linkable

☞ Top of Flood Pool

Type: Scalar

Units: No Units

Description: Operating level (as defined in Operating Level Table) corresponding to the top of the flood pool.

Information: Values must be identical for all reservoirs in the computational subbasin on which flood control is performed. May be propagated from the subbasin.

I/O: Required Input

Links: Not Linkable

☞ Top of Conservation Pool

Type: Scalar

Units: NO UNITS
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the conservation pool.
Information: Must be lower than the Top of Flood Pool. Values must be identical for all reservoirs in the computational subbasin on which flood control is performed. May be propagated from the subbasin.
I/O: Required Input
Links: Not Linkable

☞ OPERATING LEVEL TABLE

Type: Periodic
Units: TIME VS LENGTH AT OPERATING LEVELS
Description: Describes the seasonal variation of elevation (storage) in a reservoir at each of the user-designated operating levels.
Information: Number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep. See HERE (Section 22.1.19) for a method to modify which operating level table is used within a run.
I/O: Required Input
Links: Not Linkable

☞ ALLOWABLE RISING RELEASE CHANGE

Type: Scalar
Units: FLOW PER TIME
Description: The maximum acceleration of the flood control releases, used by the rules-based Operating Level Balancing flood control. Value must be greater than 0.
Information: The Operating Level Balancing flood control algorithm attempts not to exceed this increase in outflow from one timestep to the next when computing flood control releases. Surcharge releases may cause this change to be exceeded. Value is converted to the internal units appropriate to the timestep size of the run, thus, is independent of the run’s timestep size.
I/O: Required Input
Links: Not Linkable

☞ ALLOWABLE FALLING RELEASE CHANGE

Type: Scalar

Units: Flow Per Time
Description: The maximum deceleration of the flood control releases, used by the rules-based Operating Level Balancing flood control. Value must be greater than 0.
Information: The Operating Level Balancing flood control algorithm attempts not to exceed this reduction in outflow from one timestep to the next when computing flood control releases. Channel space limitations may cause this change to be exceeded. Value is converted to the internal units appropriate to the timestep size of the run, thus, is independent of the run's timestep size.
I/O: Required Input
Links: Not Linkable

☞ Maximum Release Variation

Type: Scalar
Units: Flow Per Time
Description: A deceleration of the flood control releases assumed to occur during calculation of channel space in the rules-based Operating Level Balancing flood control. Value must be greater than 0.
Information: The Operating Level Balancing flood control algorithm uses this value as a tuning parameter. Large values mean that flood releases may be larger "today" and thus, zero "tomorrow" due to routing lags, whereas small values may mean that less-than-ideal releases can be made "today"; in either case, poor choices can cause oscillations in the reservoirs' releases. Value is converted to the internal units appropriate to the timestep size of the run, thus, is independent of the run's timestep size.
I/O: Required Input
Links: Not Linkable

☞ Through Release

Type: Series
Units: Flow
Description: Water released from upstream reservoirs that flows through this reservoir. This reservoir is a downstream tandem.
Information: This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output only
Links: Not Linkable

☞ Tandem Storage

Type: Series
Units: Flow

Description: All or part of a proposed release may be stored in downstream “tandem” reservoirs if the downstream reservoir is not Surcharging (Surcharge Release equals zero or is not valid) at the current controller timestep. This slot tracks the water stored in this reservoir.
Information: This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output only
Links: Not Linkable

☞ Lost Tandem Storage
Type: Agg Series
Units: Flow
Description: This slot tracks the lost storage (as a flow) that occurs because tandem releases are not routed from the upstream to the downstream reservoir. It is really an error term. The first column is total of the other columns. Colum2 is for the current timestep releases. Column 2 is for timestep 1 through the balance period releases. Column 3 is for releases from the balance period through forecast period. For example the code decides to release 100cfs that can be stored in the downstream tandem. If you don't route it, 100 gets stored on the current timestep. But if you had routed it, some fraction would not have made it there on day 1 (1st column). If you instead propose a hydrograph of flows that don't get routed, then you can compute the col 2 and col 3 values.
Information: This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output Only

☞ Downstream Control Point Share
Type: Agg Series
Units: DECIMAL
Description: The slot has a series slot for each downstream key control point. The value in the slot shows the share as presented on the downstream Control Point’s Share slot. Essentially, this slot contains a copy of the downstream Control Point’s Share information.
Information: This slot is invisible but can be viewed in the reservoir’s Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output only
Links: Not Linkable

24.1.18.3 Phase Balancing

Choose this method when you wish to use rules to perform Phase Balancing flood control on a computational subbasin of which this reservoir is a member. See Phase Balancing discussion in the Computational Object for a complete discussion of the Phase Balancing algorithm.

Slots Specific to this Method

☞ **Forecast period**
Type: Scalar
Units: No Units
Description: Number of timesteps in the forecast period.
Information: Minimum value of 1. This value should be the same as the Forecast Period on other objects in the subbasin; the value can optionally be propagated from the subbasin.
I/O: Required Input
Links: Not Linkable

☞ **Surcharge Release**
Type: Series
Units: Flow
Description: release required by operating policy when the reservoir elevation is in the surcharge pool.
Information: set by surcharge release method for all timesteps in forecast period when “Surcharge Release” flag is set on this slot by end user or by a rule
I/O: Output
Links: Not Linkable

☞ **Top of Flood Pool**
Type: Scalar
Units: No Units
Description: operating level (as defined in Operating Level Table) corresponding to the top of flood pool
Information: KC COE has used 4 in the past, but can be any number
I/O: Required Input
Links: Not Linkable

☞ **Top of Conservation Pool**
Type: Scalar
Units: None

Description: The operating level pool number of the top of conservation pool level.
Information: This value is also termed the Target Operating Pool Level, since this is the preferred, or "Target" level for all reservoirs. This level is also the bottom of the flood pool level.
I/O: Input
Links: Not linkable

☞ Operating Levels Table

Type: Periodic
Units: TIME vs Length at Operating levels
Description: table describing the seasonal variation of elevation (storage) in a reservoir at each of the user-designated operating levels.
Information: number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep. See HERE (Section 22.1.19) for a method to modify which operating level table is used within a run.
I/O: Required Input
Links: Not Linkable

☞ Operating Level

Units: Series
Units: No units
Description: Operating level of the reservoir at each timestep, based on the ending Storage.
Information: Determined from interpolation of the Operating Levels Periodic plus the delta from the Operating Levels Aberration Agg Series Slot; an exact value is determined, not limited to an integer or values entered in the Operating Level Table.
I/O: Output
Links: Not Linkable

☞ Phase Tolerance

Type: Scalar
Units: No Units
Description: The tolerance value that must be exceed for the reservoir to change phase.
Information: The reservoir phase is detirmined from the operating level. The operating level must exceed a phase boundary by the "Phase Tolerance" before the lake

will change phase. This slot can be used to smooth oscillations along phase boundaries.

I/O: Input (0.0 default)

Links: Not Linkable

☞ Lake Character

Type: Series

Units: No units

Description: weighting factor for allocating available downstream channel space.

Information: used by control point or flood control algorithm to balance reservoirs in calculating flood releases

I/O: Output

Links: Not Linkable

☞ Lake Character Coefficient

Type: Scalar

Units: No UNITS

Description: coefficient multiplied by the occupied volume of the flood pool to determine the Lake Character.

Information: used for every timestep, unless Variable Lake Character Coefficient has been input (not a NaN) for some timestep(s).

I/O: Input

Links: Not Linkable

☞ Variable Lake Character Coefficient

Type: Series

Units: NO units

Description: time-varying Lake Character Coefficient

Information: replaces Lake Character Coefficient Scalar when input for some timestep(s).

I/O: Optional Input

Links: Not Linkable

☞ Flood Control Release

Type: Series

Units: FLOW

Description: value of the flood control release calculated by the Flood Control Release method and set by the rule that calls that method.

Information: During the flood control calculations at each timestep, this value is calculated for the entire forecast period; at the next timestep, the forecast values are overwritten by the new values. At the end of the run, this slot contains the flood control release for all the timesteps. During dispatching, this value is added to the surcharge release and possibly other values to get the total outflow for the timestep.

I/O: Output

Links: Not linkable

☞ PROPOSED FLOOD CONTROL RELEASE

Type: Series
Units: FLOW
Description: The running value of the flood control release calculated by the Flood Control Release method on the Computation Subbasin.
Information: This slot is used to hold "proposed" flood control release during the flood control calculations. This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output
Links: Not linkable

☞ RECALC RELEASE

Type: Series
Units: FLOW
Description: The temporary running value of the reservoir's current constrained release.
Information: This slot is used to hold the last constrrained release of the reservoir. This value is kept in case the computational subbasin has the opportunity to recalculate the reservoir's release (i.e., one of this reservoir's siblings cannot use all of its allocation allowing this flood control release of this reservoir to be increased). This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output
Links: Not linkable

☞ KNOWN RELEASE

Type: Series
Units: FLOW
Description: The temporary running value of the reservoir's "known" or solved for releases.
Information: This slot is used for bookkeeping, holding the values of "known" or solved for releases in the forecast period during the flood control release algorithm. This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output
Links: Not linkable

☞ Phase

Type: Series
Units: No Units
Description: A temporary record of the lake's phase.
Information: This slot is used for bookkeeping, holding the values of the lake phase. This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output
Links: Not Linkable

☞ Minimum Release

Type: Periodic
Units: FLOW
Description: The minimum release the reservoir is required to make. A seasonal value.
Information: number of rows defined by the number of date points (user input) and one column. The column represents the time-varying minimum release for the reservoir.elevations for a particular Operating Level. An flow value is input for each date point. User can select whether to interpolate between values in time, or to have constant values until the next timestep.
I/O: Required Input
Links: Not Linkable

☞ Tandem Operating Levels Table

Type: Periodic
Units: TIME vs Length at Operating levels
Description: Table describing the seasonal variation of elevation (storage) in a tandem reservoir at each of the user-designated operating levels.
Information: number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep.
I/O: Required Input
Links: Not Linkable

☞ Tandem Operating Level Aberrations

Type: Agg Series Slot
Units: No units

Description: Aberration of the operating levels of the tandem reservoir from the periodic specification at each timestep
Information: The slot contains an aberration series for each operating level. The abberation is used to adjust the tandem operating levels when constructing the tandem balancing curve.
I/O: Input
Links: Not Linkable

☞ Objective Release Pattern Table

Type: Table
Units: No Units
Description: Table describing the objective releases percentages used to evacuate the flood control storage (including forecasted inflows) over the period of the table from the first unconstrained release.
Information: The n^{th} number in the table represents the percentage of the flood control storage (include forecasted inflows) that should be released for the n^{th} day from the first unconstrained release from this reservoir.
I/O: Required Input
Links: Not Linkable

☞ Maximum Objective Release

Type: Series
Units: FLOW
Description: The temporary running values of the reservoir's current objective releases. This slot is invisible but can be viewed in the Special Results details on the Model Run Analysis tool HERE (USACE_SWD.pdf, Section 5.5).
I/O: Output
Links: Not linkable

☞ Objective Pattern Threshold

Type: Table
Units: NONE
Description: Percentage of allowable volume change in objective release pattern
Information: This is a Scalar representing the percentage of volume changed. If the change volume of flood control storage in a reservoir (including forecasted inflows) is within the threshold of the objective release pattern, then the pattern can be maintained and does not need to be recomputed. If the change exceeds the threshold then the pattern must be recomputed. The table has two columns. The first column lists outflows. The second column list the corresponding permissible decrease in outflow.
I/O: Required Input

Links: Not Linkable

☞ PERMISSIBLE OUTFLOW INCREASE CONSTRAINTS TABLE

Type: Table
Units: FLOW, FLOW
Description: Table describing the allowable outflow increase given a current outflow.
Information: The table has two columns. The first column lists outflows. The second column list the corresponding permissible increase in outflow.
I/O: Required Input
Links: Not Linkable

☞ PERMISSIBLE OUTFLOW DECREASE CONSTRAINTS TABLE

Type: Table
Units: FLOW, FLOW
Description: Table describing the allowable outflow decrease given a current outflow.
Information: The table has two columns. The first column lists outflows. The second column list the corresponding permissible decrease in outflow.
I/O: Required Input
Links: Not Linkable

24.1.19 Bank Storage

The Bank Storage methods are used to calculate the volume of water stored in the Reservoir banks. These methods also calculate the change in the volume of water stored in the Reservoir banks from one timestep to the next.

24.1.19.1 None

None should be chosen if the user does not want to calculate the amount of Bank Storage in the Reservoir. This is the default method for the Bank Storage category. Bank Storage and the Change in Bank Storage are set to zero but are not displayed. There are no slots specifically associated with this method. No calculators are performed in this method.

24.1.19.2 Input Bank Storage

The Input Bank Storage method allows users to directly input values into the Bank Storage slot or to set these values using a rule. Change in Bank Storage is calculated internally in RiverWare for use in the mass balance equations.

Slots Specific to this Method

☞ **Bank Storage**

Type: Series Slot
Units: Volume
Description: volume of water stored in the reservoir banks
Information:
I/O: Input Only
Links: Usually not linked, but could be linked to Data Object.

☞ **Change in Bank Storage**

Type: Series Slot
Units: Volume
Description: change in volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

24.1.19.3 CRSS Bank Storage

The CRSS Bank Storage method replicates the U.S. Bureau of Reclamation's CRSS bank storage calculation. The Bank Storage and the Change in Bank Storage are calculated using the Reservoir Storage and the Bank Storage Coefficients.

SLOTS SPECIFIC TO THIS METHOD

☞ BANK STORAGE

Type: Series Slot
Units: VOLUME
Description: volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

☞ BANK STORAGE COEFFICIENT

Type: Table Slot
Units: NO UNITS VS. NO UNITS
Description: gain or loss of storage vs. change in bank storage
Information: The first coefficient (column zero) is for increasing storage and the second coefficient is for decreasing storage.
I/O: Required input
Links: Not linkable

☞ CHANGE IN BANK STORAGE

Type: Series Slot
Units: VOLUME
Description: change in volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

There are two ways Bank Storage can be calculated depending on the current Storage of the Reservoir. If the Reservoir's current Storage is greater than the Reservoir's Storage at the previous timestep, the Storage is increasing. Bank Storage is calculated using the following equation:

$$\text{Bank Storage} = \text{Bank Storage(-1)} + (\text{first Bank Storage Coefficient} \times (\text{Storage} - \text{Storage(-1)}))$$

If the Reservoir's current Storage is less than the Reservoir's Storage at the previous timestep, the Storage is decreasing. Bank Storage is calculated using the following equation:

$$\text{Bank Storage} = \text{Bank Storage(-1)} + (\text{second Bank Storage Coefficient} \times (\text{Storage} - \text{Storage(-1)}))$$

The Change in Bank Storage is calculated using the following equation regardless of which method was used to compute Bank Storage.

$$\text{Change in Bank Storage} = \text{Bank Storage} - \text{Bank Storage}(-1)$$

where in the above equations:

Bank Storage = the volume of water stored in the banks of the Reservoir at the current timestep

Bank Storage(-1) = the volume of the water stored in the banks of the Reservoir at the previous timestep.

Storage = the volume of water in the Reservoir at the current timestep

Storage(-1) = the volume of water in the Reservoir at the previous timestep

24.1.19.4 Average Stage Change

The Average Stage Change method calculates the Bank Storage and Change in Bank Storage based on the flow from storage. The flow from storage is a function of the average stage change over a user defined number of timesteps.

Slots Specific to this Method

Ave Stage Change Coeffs

Type: Table Slot
Units: Area per Time and Flow
Description: coefficient describing flow for a given change in pool elevation and a constant representing flow from bank storage
Information:
I/O: Required input
Links: Not linkable

Bank Storage

Type: Series Slot
Units: Volume
Description: volume of water stored in the reservoir banks
Information:
I/O: Output only
Links: Not linkable

Change in Bank Storage

Type: Series Slot
Units: Volume
Description: change in volume of water stored in the reservoir banks
Information:

I/O: Output only

Links: Not linkable

☞ Timesteps to Average

Type: Table Slot
Units: No Units
Description: number of timesteps used to calculate average pool elevation.
Information:
I/O: Required input

Links: Not linkable

The average stage change is calculated using the following equation:

$$\text{average Pool Elevation} = \frac{\text{Pool Elevation} - \text{Pool Elevation (-Timesteps to Average)}}{\text{Timesteps to Average}}$$

The change in flow to bank storage is calculated using the following equation:

$$\text{Flow to banks} = \text{Average Stage Change Bank Storage Coefficient} \times \text{Average Pool Elevation} + \text{Average Stage Change Bank Storage Constant}$$

The flow is converted to a volume by multiplying the value by the current timestep. The Change in Bank Storage is calculated using the following equation:

$$\text{Change in Bank Storage} = \text{Bank Storage} - \text{Bank Storage(-1)}$$

24.1.20 Diversion from Reservoir

The Diversion from Reservoir user methods are applicable when a reservoir is linked to a diverting object (e.g. AggDiversionSite, AggDistributionCanal, or Diversion Object). These methods simply create the slots which must be linked (by the user) to slots on the diverting object.

24.1.20.1 None

This is the default for the Diversion from Reservoir category. It is used when the reservoir is not linked to a diverting object. If the reservoir is linked to a diverting object and this method is selected, the object will not solve correctly. There are no slots specifically associated with this method.

24.1.20.2 Available Flow Based Diversion

This method must be selected when a reservoir is linked to either an AggDiversionSite, AggDistributionCanal, or a Diversion Object that is using the Available For Diversion Linked method. Selecting this method allows the Available for Diversion slot to be available for linking. The AggDiversionSite, AggDistributionCanal, and Diversion objects contain more information about diverting water from a reservoir.

Slots Specific to this Method

☞ **Available for Diversion**

Type: Series
Units: Flow
Description: represents the amount of water that may be diverted from the reservoir
Information:
I/O: Optional; can be input by the user or determined by **RiverWare**™.
Links: Should be linked to the Available for Diversion slot on AggDiversionSite or Diversion object, or the Incoming Available Water slot on a Water User.

Available for Diversion can either be input by the user or calculated by the reservoir. If it is not input it is set as the previous Storage divided by the timestep length. The value is limited to not be negative.

No other calculations are performed if this method is selected.

24.1.20.3 Head Based Diversion

This method may be selected when a reservoir is linked to a Diversion Object. Selecting this method allows the Previous Pool Elevation slot to be available for linking. The Diversion Object contains more information about diverting water from a Reservoir.

Slots Specific to this Method

☞ Previous Pool Elevation

Type: Series
Units: LENGTH
Description: Pool Elevation value for the previous timestep
Information:
I/O: Output only
Links: Should be linked to the Diversion Intake Elevation slot on the Diversion Object.

24.1.21 Diversion Power

The methods in this category calculate power generated on the diversion from the reservoir. Selecting a method other than None in this category will make the Diversion Tailwater and Diversion Power Bypass categories available. If a method other than None is selected for Diversion Power, then a method other than None must be selected for Diversion Tailwater.

24.1.21.1 None

This is the default method for the Diversion Power category. No calculations are performed in this method, and there are no slots specifically associated with this method.

24.1.21.2 Diversion Power Efficiency Curve

The Diversion Power Efficiency Curve method is similar to the Plant Efficiency Curve method in the Power category with the exception that the method does not allow **Diversion Energy** to input or set by rules (nor can **Diversion Energy** be set with the Best Efficiency or Max Capacity flags). **Diversion Energy** and **Diversion Power** are only calculated as outputs. The method calculates **Diversion Power** by a 3-D interpolation of the **Diversion Power Table** using the current, average **Diversion Operating Head** and **Diversion Turbine Flow**. The **Diversion Power Coefficient** is calculated as **Diversion Power** divided by **Diversion Turbine Flow**. Alternatively, the user can input **Diversion Power Coefficient**, and then **Diversion Power** is calculated directly as the **Diversion Power Coefficient** multiplied by the **Diversion Turbine Flow**.

SLOTS SPECIFIC TO THIS METHOD

☞ **DIVERSION POWER TABLE**

Type: Table Slot
Units: LENGTH VS FLOW VS POWER
Description: 3-D table representing the power characteristics of the diversion power plant, used to calculate power using interpolation
Information: Data must be entered into the table in increasing, blocks of the same Diversion Operating Head. For every block of the same Diversion Operating Head in column 1, Diversion Turbine Flow should be listed in increasing order in column 2 and the corresponding Diversion Power in column 3. The first row for each Diversion Operating Head must be for zero Diversion Turbine Flow and zero Diversion Power.
I/O: Required input
Links: Not linkable

Div Head	Turbine Flow	Div Power
30	0	0
30	100	100
30	200	175
40	0	0
40	100	125
40	220	195
50	0	0
50	110	147
50	250	205

☞ Diversion Max Turbine Table

Type: Table Slot
Units: Length vs Flow
Description: The maximum Diversion Turbine Flow as a function of Diversion Operating Head
Information: RiverWare automatically populates this table at the start of the run using the Diversion Power Table. The first column contains the Diversion Operating Head values from the Diversion Power Table, one row for each unique Diversion Operating Head in increasing order. The second column contains the maximum Diversion Turbine Flow value for each Diversion Operating Head.
I/O: Output only
Links: Not Linkable

☞ Diversion Power Cap Fraction

Type: Series Slot
Units: Fraction
Description: This is the percentage of full capacity of the turbine units in the diversion power plant. For example, if only half of the turbine are operational (and they are all the same), this value would be 0.5.
Information: This must be a number between 0 and 1 (inclusive). If not input or set by rules, this slot is automatically set to 1.
I/O: Optional input, if not, value is set to 1
Links: Not linkable

☞ Diversion Operating Head

Type: Series Slot

Units: LENGTH
Description: The difference between the average Pool Elevation and the Diversion Tailwater Elevation
Information:
I/O: Output only
Links: Not usually linked

☞ DIVERSION TURBINE FLOW

Type: Series Slot
Units: FLOW
Description: The diversion flow that passes through the turbines to generate power
Information: If the slot is not input or set by rules, then it is calculated as the difference between Diversion and Diversion Power Bypass if Diversion Power Bypass is input or set by rules. If neither Diversion Turbine Flow nor Diversion Power Bypass is input or set by rules, then Diversion Turbine Flow is calculated as the lesser of Diversion and the calculated maximum diversion turbine flow based on the Diversion Max Turbine Table and the current Diversion Operating Head. It is not permissible to have both Diversion Turbine Flow and Diversion Power Bypass as input or set by rules.
I/O: Optional input or output
Links: Not linkable

☞ DIVERSION POWER

Type: Series Slot
Units: POWER
Description: The power generated from flow through the reservoir diversion
Information: If Diversion Power Coefficient is not input or set by rules, Diversion Power is calculated using a 3-D interpolation on the Diversion Power Table given the current, average Diversion Operating Head and the current Diversion Turbine Flow, scaled by the Diversion Power Cap Fraction. If Diversion Power Coefficient is input or set by rules, Diversion Power is calculated as the Diversion Power Coefficient multiplied by the Diversion Turbine Flow.
I/O: Output only
Links: Linkable

☞ DIVERSION ENERGY

Type: Series Slot
Units: ENERGY
Description: The energy generated from flow through the reservoir diversion
Information: Calculated as the Diversion Power multiplied by the timestep length
I/O: Output only

Links: Linkable

☞ **DIVERSION POWER COEFFICIENT**

Type: Series Slot
Units: POWER PER FLOW
Description: The power generation per unit of flow through the turbines on the reservoir diversion
Information: If this slot is input or set by rules, it is used directly to calculate Diversion Power. If it is not input or set by rules, then it is calculated as Diversion Power divided by Diversion Turbine Flow. If either Diversion Power, Diversion Turbine Flow or Diversion Power Cap Fraction is zero, then this slot will be zero.
I/O: Optional input or output
Links: Not usually linked

At the start of the run, the **Diversion Max Turbine Table** slot is populated using the **Diversion Power Table**. The first column is populated with each unique **Diversion Operating Head** value from the **Diversion Power Table**, in ascending order. The second column is populated with the corresponding maximum diversion turbine flow value.

When the method executes, **Diversion** will already be known. The method calls the selected Diversion Tailwater method to calculate the **Diversion Tailwater Elevation**. If the default method, **None**, is selected for the **Diversion Tailwater** category, the run will abort and an error message will be issued.

The method then calculates the **Diversion Operating Head**:

$$\text{Diversion Operating Head}[t] = \frac{\text{Pool Elevation}[t-1] + \text{Pool Elevation}[t]}{2} - \text{Diversion Tailwater Elevation}[t]$$

The method calculates the *maxDiversionTurbineFlow* by interpolating the **Diversion Max Turbine Table** slot using the **Diversion Operating Head**. The value is scaled by the **Diversion Power Cap Fraction**.

If **Diversion Turbine Flow** is specified (input or set by rules) it is checked against the *maxDiversionTurbineFlow*, and if the specified value exceeds the max, the run will abort with an error message. Otherwise a temporary turbine flow is calculated.

$$\text{tempDiversionTurbine} = Min(\text{Diversion} - \text{Diversion Power Bypass}, maxDiversionTurbineFlow)$$

If **Diversion Power Bypass** is not input or set by rules, or if the **Diversion Power Bypass** method is **None**, the **Diversion Power Bypass** will have defaulted to zero at this point.

If the combined temporary turbine flow plus the current **Diversion Power Bypass** is less than the (total) **Diversion**, then the method then calls the selected Diversion Power Bypass method to increase the **Diversion Power Bypass** to make up the difference. If it is not possible for the turbine flow plus the bypass to equal the total **Diversion**, either due to the values being specified (input or rules) or due to max capacity limits, then the run will abort with an error message.

The method then sets the Diversion Turbine Flow slot:

$$\text{Diversion Turbine Flow} = \text{tempDiversionTurbine}$$

If **Diversion Power Coefficient** is specified (input or rules), it is used to calculate **Diversion Power** directly:

$$\text{Diversion Power} = \text{Diversion Turbine Flow} \times \text{Diversion Power Coefficient}$$

Otherwise, **Diversion Power** is calculated by a 3-D interpolation on the **Diversion Power Table** using **Diversion Operating Head** and **Diversion Turbine Flow**, and **Diversion Power Coefficient** is calculated as:

$$\text{Diversion Power Coefficient} = \frac{\text{Diversion Power}}{\text{Diversion Turbine Flow}}$$

Diversion Energy is then calculated as **Diversion Power** multiplied by the timestep length.

Notes on Diversion Power Cap Fraction

If the **Diversion Power Cap Fraction** is input by the user, it is necessary for the **Diversion Power Table** to be scaled back to account for the operating points when the turbines are operating at less than 100%. To do this, **Diversion Turbine Flow** is divided by the **Diversion Power Cap Fraction**. This point is then found in the **Diversion Power Curve** for the current **Diversion Operating Head**, and the power is found using 3-D interpolation. Finally the power is multiplied by the **Diversion Power Cap Fraction** to get the actual **Diversion Power** produced for the current timestep.

24.1.22 Diversion Tailwater

The methods in this category calculate the elevation of the tailwater on the diversion from a reservoir. This category is dependent on the selection of a method other than the default method, None, in the Diversion Power category. If a method other than the default is selected for Diversion Power, then a method other than the default, None, must be selected for Diversion Tailwater.

24.1.22.1 None

This is the default method for the Diversion Tailwater category. No calculations are performed in this method, and there are no slots specifically associated with this method.

24.1.22.2 Diversion Base Value Plus Lookup

The Diversion Base Value Plus Lookup method computes the **Diversion Tailwater Elevation** by added the average **Diversion Tailwater Base Value** (over the timestep) to a function of **Diversion** defined in the **Diversion Tailwater Table** slot. This method is similar to the Base Value Plus Lookup Table method in the Tailwater category but uses the **Diversion** and **Diversion Tailwater Base Value** slots instead of **Outflow** and **Tailwater Base Value**. The **Diversion Tailwater Base Value** may be input by the user or linked to another slot, such as the **Pool Elevation** of another Reservoir. If the **Tailwater Base Value** is neither input nor linked, it is automatically set to zero.

Slots Specific to this Method

☞ **Diversion Tailwater Table**

Type: Table Slot
Units: Flow vs Length
Description: This slot defines the relationship between Diversion and the Diversion Tailwater Elevation; Diversion vs either the diversion tailwater elevation or the tailwater elevation increment
Information: If the Diversion Tailwater Base Value is non-zero, the Diversion Tailwater Table gives values of incremental increase in Tailwater Elevation over th base value. Otherwise, the table gives the Diversion Tailwater Elevation values. The first row of the table should be for a Diversion flow of zero.
I/O: Required input
Links: Not linkable

☞ **Diversion Tailwater Base Value**

Type: Series Slot
Units: Length
Description: the base elevation of the diversion tailwater, such as a downstream stage

Information: If the slot is not input or linked, it defaults to 0.
I/O: Optional, can be input or linked
Links: Linkable

☞ DIVERSION TAILWATER ELEVATION

Type: Series Slot
Units: LENGTH
Description: the water surface elevation of the tailwater from the reservoir diversion
Information: This slot is used to compute Diversion Operating Head in Diversion Power calculations
I/O: Output only
Links: Not linkable

When this method is executed, the **Diversion** value will already be known. If the **Diversion Tailwater Base Value** is neither linked, input nor set by rules, then it will default to zero.

The following steps are performed to calculate **Diversion Tailwater Elevation**.

1. *TWTemp* is obtained from a table interpolation on the **Diversion Tailwater Table** using **Diversion**.

2. If both **Diversion Tailwater Base Value[t]** and **Diversion Tailwater Base Value[t-1]** are known, then the **Diversion Tailwater Elevation** is calculated as:

$$\text{Diversion Tailwater Elevation}[t] = \frac{\text{Diversion Tailwater Base Value}[t-1] + \text{Diversion Tailwater Base Value}[t]}{2} + TWTemp$$

3. If **Diversion Tailwater Base Value[t]** is known, but **Diversion Tailwater Base Value[t-1]** is not known, then the **Diversion Tailwater Elevation** is calculated as:

$$\text{Diversion Tailwater Elevation}[t] = \text{Diversion Tailwater Base Value}[t] + TWTemp$$

4. If **Diversion Tailwater Base Value[t-1]** is known, but **Diversion Tailwater Base Value[t]** is not known, then the **Diversion Tailwater Elevation** is calculated as:

$$\text{Diversion Tailwater Elevation}[t] = \text{Diversion Tailwater Base Value}[t-1] + TWTemp$$

5. If neither **Diversion Tailwater Base Value[t]** nor **Diversion Tailwater Base Value[t-1]** are known but **Diversion Tailwater Elevation[t-1]** is known, the current timestep's **Diversion Tailwater Elevation** is set equal to **Diversion Tailwater Elevation[t-1]**.

6. If neither **Diversion Tailwater Base Value[t]**, **Diversion Tailwater Base Value[t-1]**, nor **Diversion Tailwater Elevation[t-1]** are known, or if **Diversion** is not known, the method will exit and wait for more information.

24.1.23 Diversion Power Bypass

The methods in this category calculate the portion of the diversion from a reservoir that does not pass through the turbines but rather through a bypass structure. This category is dependent on the selection of a method other than the default method, None, in the Diversion Power category.

24.1.23.1 None

This is the default method for the Diversion Power Bypass category. No calculations are performed in this method, and there are no slots specifically associated with this method. If this method is selected, it is assumed that all **Diversion** flow passes through the turbines.

24.1.23.2 Bypass Capacity Table

This method sets **Diversion Power Bypass** to the difference between **Diversion** and **Diversion Turbine Flow** if it is not input or set by rules, and it checks that the **Diversion Power Bypass** does not exceed the maximum based on the **Diversion Power Bypass Table**. This functions similarly to the Regulated method in the Spill category.

Slots Specific to this Method

☞ **Diversion Power Bypass Table**

Type: Table Slot
Units: Length vs Flow
Description: Pool Elevation vs. the corresponding maximum diversion power bypass values
Information:
I/O: Required input
Links: Not linkable

☞ **Diversion Power Bypass**

Type: Series
Units: Flow
Description: Diversion flow that does not pass through power turbines
Information: If not input or set by rules, Diversion Power Bypass will be set equal to the difference between Diversion and Diversion Turbine Flow.
I/O: Optional input or output
Links: Linkable

At the beginning of the run, if **Diversion Power Bypass** is not specified (input or rules), it is initially set to a default value of zero.

On each timestep, the method first checks if both **Diversion Power Bypass** and **Diversion Turbine Flow** are input or set by rules. If so, the run will abort with an error message. It necessary to leave at least one of these slots as a free variable.

The method then calculates max diversion power bypass by performing a table interpolation on the **Diversion Power Bypass Table** using the average **Pool Elevation** from the end of the current timestep and end of the previous timestep.

If **Diversion Power Bypass** is input or set by a rule, then the value is checked against the max diversion power bypass, and if it exceeds the max, the run will abort with an error message.

If **Diversion Power Bypass** is not input or set by a rule, then it is calculated as:

$$\text{Diversion Power Bypass} = \text{Diversion} - \text{Diversion Turbine Flow}$$

The calculated **Diversion Power Bypass** value is checked against the max diversion power bypass, and if it exceeds the max, the run will abort with an error message.

24.1.24 Seepage

The Seepage methods are used to calculate the amount of water lost through the face of the dam. The volume of seepage computed during the execution of these methods affects the mass balance of the Reservoir.

24.1.24.1 None

None is the default for the Seepage category. It is used when the user does not want to calculate the flow of water through the face of the dam.

24.1.24.2 Input Seepage

The Input Seepage method is used when it is desired to have the seepage slot as input or set by a rule.

SLOTS ADDED BY THIS METHOD:

☞ **SEEPAGE**

Type: SeriesSlot
Units: FLOW
Description: flow of water through the dam face
Information: Seepage is not included in the Outflow of the reservoir and will need to be linked separately if the water does in fact go downstream.
I/O: Input only
Links: Linkable

24.1.24.3 Linear Seepage

The Linear Seepage method calculates the seepage from the face of the dam. This calculation is based on the Pool Elevation of the Reservoir and specified coefficients.

SLOTS ADDED BY THIS METHOD:

☞ **SEEPAGE**

Type: SeriesSlot
Units: FLOW
Description: flow of water through the dam face
Information: Seepage is not included in the Outflow of the reservoir and will need to be linked separately if the water does in fact go downstream.
I/O: Output only
Links: Linkable

☞ **SEEPAGE COEFFICIENTS**

Type: TableSlot
Units: LENGTH, AREA PER TIME, FLOW
Description: coefficients in the linear equation for seepage
Information: The first coefficient (column zero) is the base elevation of the dam. The second coefficient is the slope of the linear equation for seepage. The third coefficient is the intercept of the linear equation for seepage.
I/O: Required Input
Links: NA

The calculation for Seepage in this method is fairly straightforward. A linear model is used. The coefficient are user inputs. The following equation is used to compute Seepage:

$$\text{Seepage} = (\text{Pool Elevation} - \text{first Seepage Coefficient}) \times \text{second Seepage Coefficient} + \text{third Seepage Coefficient}$$

24.1.24.4 Single Seepage Value

The Single Seepage Value method sets the seepage from the face of the dam equal to a scalar value.

SLOTS ADDED BY THIS METHOD:

☞ **SEEPAGE**

Type: Series
Units: FLOW
Description: Flow of water through the dam face
Information: Seepage is not included in the Outflow of the reservoir and will need to be linked separately if the water does in fact go downstream.
I/O: Optional Input
Links: Linkable

☞ **SINGLE SEEPAGE VALUE**

Type: Scalar
Units: FLOW
Description: seepage value to be applied to each timestep
Information:
I/O: Required Input
Links: Not Linkable

This method is executed at the beginning of the run. For each timestep from the initial timestep through the end of the run (plus post run dispatching timesteps too), if the **Seepage** is not input, the **Seepage** is set equal to the **Single Seepage Value**.

The method will issue an error if there is not a valid value in the **Single Seepage Value** slot. Note, this structure allows some flexibility. Seepage can be input/rules when necessary but will use the scalar value when not input.

24.1.24.5 Linked Seepage

This method is intended to be used when linking a Reservoir object with a Groundwater Storage object that uses the **Head Based Boundary Condition** method in the **Solution Type** category **HERE (Objects.pdf, Section 14.1.1.3)**.

SLOTS ADDED BY THIS METHOD:

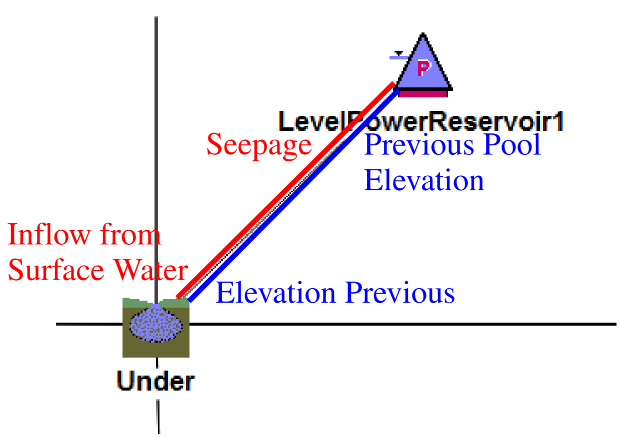
☞ **SEEPAGE**

Type: Series
Units: FLOW
Description: Flow of water out of the reservoir, often into groundwater
Information: A positive value is flow out of the reservoir.
I/O: Output only if linked to a Groundwater object (typical); otherwise required input
Links: Must be linked, typically to **Inflow from Surface Water** on Groundwater object

☞ **PREVIOUS POOL ELEVATION**

Type: Series
Units: LENGTH
Description: Pool Elevation at the end of the previous timestep
Information:
I/O: Output only
Links: Typically linked to the **Elevation Previous** slot on a Groundwater object

This method does not do any calculations; it just adds the appropriate slots. The Reservoir does provide the **Previous Pool Elevation** which is then linked to the Groundwater object **Elevation Previous**. The Groundwater object computes the **Inflow from Surface Water**, which is linked back to the **Seepage** slot on the reservoir. The **Seepage** is used in the Reservoir mass balance.



Click HERE (Objects.pdf, Section 14.1.1.3) for more information about how **Seepage** is calculated as **Inflow from Surface Water** on the linked Groundwater object.

24.1.25 Operating Levels

This category enables the user to specify operating levels for the reservoir. Operating levels serve as a normalizing metric for reservoir contents. This metric is used by reservoir-balancing algorithms to determine the relative “fullness” of reservoirs. On individual reservoirs, it also serves to identify the elevations that correspond to pool boundaries, such as the top of the conservation pool or the top of the flood pool.

24.1.25.1 None

This is the default method; no slots are instantiated and no calculations are performed.

24.1.25.2 Conservation Pools

This method allows the user to specify that there is a conservation pool for this reservoir.

SLOTS SPECIFIC TO THIS METHOD

☞ **OPERATING LEVEL TABLE**

Type: PeriodicSlot
Units: TIME VS LENGTH AT OPERATING LEVELS
Description: table describing the seasonal variation of elevation in a reservoir at each of the user-designated operating levels.
Information: number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep.
I/O: Required Input
Links: Not Linkable

☞ **OPERATING LEVEL**

Type: SeriesSlot
Units: NONE
Description: The computed operating level
Information: This slot is computed using the pool elevation and the Operating Level Table
I/O: Output only
Links: Not Linkable

☞ Operating Level Storage Table

Type: PeriodicSlot
Units: Time vs Volume at Operating Levels
Description: table describing the seasonal variation of storage in a reservoir at each of the user-designated operating levels.
Information: Each column represents the time-varying storage for a particular Operating Level. The integer value of the Operating Level is in the header of each column. This table is generated from the Operating Level Table and has the same number of rows and columns. The values in the table are storage values (looked up from the elevation volume table) whereas the values in the Operating Level Table are elevations. This slot is computed at run-time so it is read-only to the user. All changes should be made in the Operating Level Table.
I/O: Output Only
Links: Not Linkable

☞ Top of Conservation Pool

Type: ScalarSlot
Units: NONE
Description: Operating level (as defined in Operating Level Table) corresponding to the top of the conservation pool.
Information:
I/O: Input
Links: Not Linkable

☞ Bottom of Conservation Pool

Type: ScalarSlot
Units: NONE
Description: Operating level (as defined in Operating Level Table) corresponding to the bottom of the conservation pool.
Information: Used by some conservation pool operations algorithms, along with the Top of Conservation Pool slot, to identify the volume in the conservation pool.
I/O: Input Only
Links: Not Linkable

☞ Conservation Pool Initial Empty Space

Type: SeriesSlot
Units: None
Description: The inflow required to fill the conservation pool at the beginning of timestep, based on the ending storage at the prior timestep, taking into account evaporation, precipitation, etc.

Information: This slot is computed at the beginning of the timestep; evaporation rates and other such factors that are not already defined at the beginning of timestep will not be taken into account in this computation. This slot is used by the water rights allocation solution algorithm to compute physical constraints and by storage accounts to compute allocation requests.
I/O: Output only
Links: Not Linkable

☞ CONSERVATION POOL STORAGE

Type: Series
Units: VOLUME
Description: This is the computed volume of water in the conservation pool.
Information: This value is always non-negative.
I/O: Output only
Links: Not Linkable

☞ CONSERVATION POOL FULL STORAGE

Type: Series
Units: VOLUME
Description: This is the possible volume of water that could be stored in the Conservation pool. It is computed as the Storage at the top of the conservation pool minus the storage at the bottom of the conservation pool
Information:
I/O: Output Only
Links: Not Linkable

☞ CONSERVATION POOL STORAGE FRACTION

Type: Periodic Slot
Units: FRACTION
Description: The values in the periodic slot represent the percentage of the conservation pool storage at each level (column) in the Operating Level Table.
Information: It has identical dimension including dates and levels as the Operating Level Table. This table will be populated at beginning of run. The Operating Level Table will be its "source" slot.
I/O: Output only
Links: Not Linkable

At the beginning of run, the Conservation Pool Storage Fraction is populated as follows: For each date (row) and each level, n, (column), the equation to compute the fraction:

$$\text{Conservation Pool Storage Fraction}[date,n] =$$

$$\frac{\text{Op Level Storage Table}[date, n] - \text{Op Level Storage Table}[date, \text{Bottom of Cons Pool}]}{\text{Op Level Storage Table}[date, \text{Top of Cons Pool}] - \text{Op Level Storage Table}[date, \text{Bottom of Cons Pool}]}$$

Note, the Conservation Pool Storage Fraction is limited to be always between 0 and 1 (0% to 100%).

At the end of each dispatch method, the Operating Level series slot is computed by looking up the pool elevation and date on the Operating Level Table.

Next, **Conservation Pool Full Storage** is calculated as follows.

$$\text{Conservation Pool Full Storage}[t] = \text{Operating Level Storage Table}[t, \text{Top of Conservation Pool}] - \text{Operating Level Storage Table}[t, \text{Bottom of Conservation Pool}]$$

The **Conservation Pool Storage** is computed as:

If the Operating Level is greater than the Top of the Conservation Pool,

$$\text{Conservation Pool Storage}[t] = \text{Full Conservation Pool Storage}[t]$$

else if the Operating Level is less than the Bottom of the Conservation Pool,

$$\text{Conservation Pool Storage}[t] = 0$$

else

$$\text{Conservation Pool Storage}[t] = \text{Storage}[t] - \text{Operating Level Storage Table}[t, \text{Bottom of Conservation Pool}]$$

24.1.25.3 Conservation and Flood Pools

This method allows the user to specify that there is a conservation and a flood pool for this reservoir.

Slots Specific to this Method

This method is an extension of the Conservation Pool method, and selecting this method causes all the slots for Conservation Pool to become available, along with the following:

☞ Top of Flood Pool

Type: ScalarSlot
Units: No Units
Description: Operating level (as defined in Operating Level Table) corresponding to the top of flood pool.

Information:
I/O: Required Input
Links: Not Linkable

☞ FLOOD POOL STORAGE

Type: Series
Units: VOLUME
Description: This is the computed volume of water in the flood pool.
Information: This value is always non-negative.
I/O: Output Only
Links: Not Linkable

☞ FLOOD POOL FULL STORAGE

Type: Series
Units: VOLUME
Description: This is the possible volume of water that could be stored in the Flood pool. It is computed as the Storage at the top of the flood pool minus the storage at the top of the conservation pool.
Information:
Links: Not Linkable

☞ FLOOD POOL STORAGE FRACTION

Type: Periodic Slot
Units: FRACTION
Description: The values in the periodic slot represent the percentage of the flood pool storage at each level (column) in the Operating Level Table.
Information: It has identical dimension including dates and levels as the Operating Level Table. This table will be populated at beginning of run. The Operating Level Table will be its "source" slot.
I/O: Output only
Links: Not Linkable

At the beginning of run, the Conservation Pool Storage Fraction is populated as described above. Then, the Flood Pool Storage Fraction is populated as follows: For each date (row) and each level, n, (column), the equation to compute the fraction is:

$$\text{Flood Pool Storage Fraction}[date,n]=$$

$$\frac{\text{Op Level Storage Table}[t, n] - \text{Op Level Storage Table}[t, \text{Top of Cons Pool}]}{\text{Op Level Storage Table}[t, \text{Top of Flood Pool}] - \text{Op Level Storage Table}[t, \text{Top of Cons Pool}]}$$

Note, the Flood Pool Storage Fraction is limited to be always greater than 0. But, it can be larger than 1 (100%). For levels above the flood pool, the percentage will be greater than 100%.

At the end of each dispatch method, the Operating Level series slot is computed by looking up the pool elevation and date on the Operating Level Table. Next, all slots associated with the Conservation Pool are computed and set as described above. Then, **Flood Pool Full Storage** is calculated as follows:

$$\text{Flood Pool Full Storage}[t] = \text{Operating Level Storage Table}[t, \text{Top of Flood Pool}] - \text{Operating Level Storage Table}[t, \text{Top of Conservation Pool}]$$

The **Flood Pool Storage** is computed as:

If the Operating Level is less than the Top of the Conservation Pool,

$$\text{Flood Pool Storage}[t] = 0$$

else

$$\text{Flood Pool Storage}[t] = \text{Storage}[t] - \text{Operating Level Storage Table}[t, \text{Top of Conservation Pool}]$$

Note, the **Flood Pool Storage** may be larger than the **Flood Pool Full Storage**. This indicates the reservoir is above the flood pool and is surcharging.

24.1.26 Conditional Operating Levels

This category provides methods that allows the user to use alternative operating level tables based on conditions in the run.

24.1.26.1 None

This is the default method; no slots are instantiated and no calculations are performed. The original Operating Level Table is used for all computations.

24.1.26.2 Sum Inflows over Interval

This method allows an alternative operating level table (i.e. a guide curve) to be used starting on a certain date if a certain combination of flows are high enough for a specified time range.

For example, if there has been a total of 200,000 acre-feet of total inflows into a specific reservoir during the months of March, April, and May, then on June 15th, the method would switch the reservoir operations to follow an alternative table. On October 15th, the reset date, the reservoir will once again use the original Operating Level Table.

Slots Specific to this Method

☞ Operating Level 2 Table

Type: Periodic
Units: Time vs Length at Operating Levels
Description: This is the alternative operating level table that is used when indicated by hydrologic conditions. This table describes the seasonal variation of elevation in a reservoir at each of the user-designated operating levels.
Information: Number of rows defined by the number of date points (user input); number of columns defined by the number of operating levels (user input). Each column represents the time-varying elevations for a particular Operating Level. The integer value of the Operating Level is in the first row (header) of each column. An elevation value is input for each operating level on each date point. All entered values have units of length. User can select whether to interpolate between values in time, or to have constant values until the next timestep. This table should have the same dimensions (rows and columns) as the Operating Level Table.
I/O: Required Input
Links: Not Linkable

☞ Operating Level 2 Trigger Volume

Type: Scalar
Units: Volume

Description: The volume of inflows (**Inflow Sum** slot) between the criteria start and finish (specified on the **Operating Level 2 Dates** slot) that causes the reservoir to use the **Operating Level 2 Table** rather than the original **Operating Level Table.**

I/O: Required Input or an error will be issued at the start of run.

Links: NA

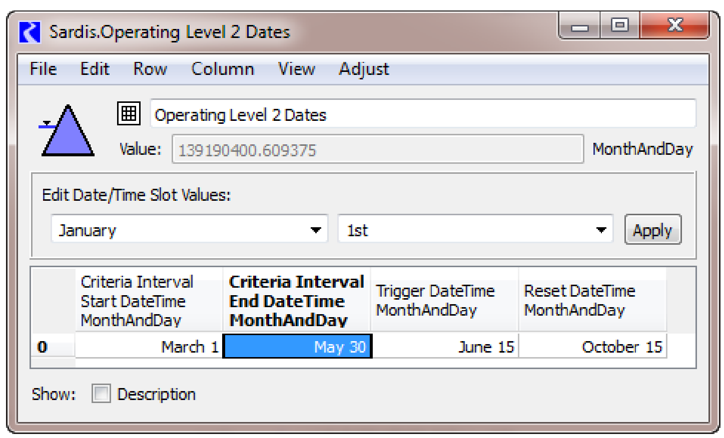
☞ Operating Level 2 Dates

Type: Table

Units: DateTime

Description: This table has 4 columns. The first and second columns are the “Criteria Interval Start DateTime” and “Criteria Interval End DateTime” between which the flow volume is summed and compared to the **Operating Level 2 Trigger Volume**. The third column is the “Trigger DateTime” at which the reservoir will check the conditions and possibly use the **Operating Level 2 Table** rather than the original **Operating Level Table.** The fourth column is the “Reset DateTime” at which the reservoir will use the **Operating Level Table** again.

Information: The units for this slot are DateTime which can be an absolute or partially specified datetime. The default user units are “MonthAndDay”. Partially specified datetimes are converted to fully specified datetimes using information from the current timestep to fill in the missing pieces of the partially specified timestep. Thus, if the datetime is partially specified, it must be able to evaluate to a timestep in the model or an error will be issued.



I/O: Required input or an error will be issued at the start of run.

Links: NA

☞ Operating Level Storage 2 Table

Type: Periodic

Units: Storage

Description: This represents the Storage associated with the elevations in the Operating Level 2 Table.

Information: This slot is created at the beginning of run. The Operating Level 2 Table is its "source" slot.
I/O: Output Only
Links: Not Linkable.

☞ Conservation Pool Storage Fraction 2

Type: Periodic Slot
Units: Fraction
Description: The values in the periodic slot represent the percentage of the conservation pool storage at each level (column) in the Operating Level Storage 2 Table.
Information: It has identical dimension including dates and levels as the Operating Level Storage 2 Table. This table will be populated at beginning of run. The Operating Level 2 Table will be its "source" slot.
I/O: Output only
Links: Not Linkable

☞ Flood Pool Storage Fraction 2

Type: Periodic Slot
Units: Fraction
Description: The values in the periodic slot represent the percentage of the flood pool storage at each level (column) in the Operating Level 2 Table.
Information: It has identical dimension including dates and levels as the Operating Level Storage 2 Table. This table will be populated at beginning of run. The Operating Level 2 Table will be its "source" slot.
I/O: Output only
Links: Not Linkable

Method Details:

This method category will be dependent on the selection of the **Conservation and Flood Pools** or **Conservation Pool** method in the **Operating Levels** method category.

At the beginning of the run, the Operating Level Storage 2 Table will be populated by looking up the elevation values in the Operating Level 2 Table on the Elevation Volume Table to get the storage associated with each level. Next the Conservation Pool Storage Fraction 2 and Flood Pool Storage Fraction 2 slots will be populated as follows:

For each date (row) and each level, n, (column), the equation to compute the fraction:

$$\text{Conservation Pool Storage Fraction 2}[date, n] =$$

$$\frac{\text{Op Level Storage 2 Table}[t, n] - \text{Op Level Storage 2 Table}[t, \text{Bottom of Cons Pool}]}{\text{Op Level Storage 2 Table}[t, \text{Top of Cons Pool}] - \text{Op Level Storage 2 Table}[t, \text{Bottom of Cons Pool}]}$$

Note, the **Conservation Pool Storage Fraction 2** is limited to be always between 0 and 1 (0% to 100%).

For each date (row) and each level, n, (column), the equation to compute the fraction:

$$\text{Flood Pool Storage Fraction}[date,n] =$$

$$\frac{\text{Op Level Storage 2 Table}[t, n] - \text{Op Level Storage 2 Table}[t, \text{Top of Cons Pool}]}{\text{Op Level Storage 2 Table}[t, \text{Top of Flood Pool}] - \text{Op Level Storage 2 Table}[t, \text{Top of Cons Pool}]}$$

Note, the Flood Pool Storage Fraction 2 is not limited to be between 0 and 1 (0% to 100%). For levels above the flood pool, the percentage will be greater than 100%.

Then, at the beginning of each timestep, the Sum Inflows over Interval method will check to see if the controller is on the “Trigger DateTime”. If so, the Inflow Sum slot will be summed (as a volume) over the criteria interval. If the sum is equal to or greater than the Operating Level 2 Trigger Volume, the reservoir will use the Operating Level 2 Table and Operating Level Storage 2 Table in all computations (until reset).

If the current timestep is a “Reset DateTime”, then the reservoir will again use the original Operating Level Table(s).

If the current timestep is neither a “Trigger DateTime” or a “Reset DateTime”, then the reservoir will reference the table used on the previous timestep. That is, it will not modify the table used but continue to use whichever table is in effect.

24.1.26.3 Cubic Bank Storage

The Cubic Bank Storage method calculates the Change in Bank Storage from an empirical equation given by the U.S. Bureau of Reclamation. This method was specifically developed for work on the Pecos. The empirical relations used in this method were derived from a study at the Reservoir. This method calculates a net overall loss rate. Bank Storage is not calculated in this method.

Slots Specific to this Method

☞ Change in Bank Storage

Type: Series
Units: VOLUME
Description: change in volume of water stored in reservoir banks
Information: Calculated by the cubic equation above.
I/O: Output only
Links: Not linkable

☞ Cubic Bank Storage Falling

Type: Table
Units: NOUNITS vs. NOUNITS
Description: constants for equation when reservoir level is falling
Information: The first coefficient (column zero) is the linear term (a) and the second coefficient is the cubic term (b).
I/O: Required input
Links: Not linkable

☞ Cubic Bank Storage Rising

Type: Table
Units: NO UNITS vs. NO UNITS
Description: constants for equation when reservoir level is rising
Information: The first coefficient (column zero) is the linear term (a) and the second coefficient is the cubic term (b).
I/O: Required input
Links: Not linkable

There are two ways the Change in Bank Storage may be calculated. Both ways of calculating the Change in Bank Storage are based on the following equation:

$$\text{Change in Bank Storage} = a + b \times \text{Pool Elevation}$$

where:

a = the first coefficient in either the Cubic Bank Storage Falling slot or the Cubic Bank Storage Rising slot

b = the second coefficient in either the Cubic Bank Storage Falling slot or the Cubic Bank Storage Rising slot

Change in Bank Storage is in cfs

Pool Elevation is in feet

Different sets of empirical coefficients are used to calculate the Change in Bank Storage depending on the direction of Pool Elevation movement. If the Pool Elevation is rising, the coefficients in the Cubic Bank Storage Rising slot are used. On the other hand, if the Pool Elevation is falling, the coefficients in the Cubic Bank Storage Falling slot are used.

24.1.27 Target Operation

The Target Operation category is used to enable algorithms which calculate different lumped mass balance algorithms required by Target Operations (see TARGET flag). None of the methods exist as separate functions, meaning that the target operation algorithms themselves must be used in conjunction with the other mass balance algorithms. It is important to note that either the Simple Target or Lagged Target method must be selected if a target operation is set on the Reservoir.

24.1.27.1 None

This is the default method in the Target Operation category. It performs no calculations. It may only be selected if a Target Operation is not performed on the object. There are no slots specifically associated with this method.

24.1.27.2 Simple Target

The Simple Target method distributes the required Inflow or Outflow evenly among all available timesteps. The total flow is divided among all the non-input timesteps included within the Target Operation to meet the Target. For most cases with a target operation, this is the desired behavior. There are no slots specifically associated with this method.

Note: Evaporation and precipitation are not included in the lumped mass balance of the target operation.

24.1.27.3 Lagged Target

The Lagged Target method should be used when there are lag times in the upstream reaches. When Inflows are known, this method solves for the same solution as the Simple Target method. When Outflows are known, this method distributes the required Inflow so a steady Outflow occurs from an upstream, non-integer timestep lagged, reservoir. To distribute Inflows, the Total Lag of Upstream Reaches and Outflow from Upstream Reservoir are used to calculate the required Inflows to the current Reservoir. These required Inflows must meet the Target and result in steady Outflow from the upstream Reservoir. The solution yields a perturbed Inflow at the first undetermined timestep of the Target Operation, followed by steady Inflows for the remainder of the undetermined target times. This solution removes the numerical instability which would be produced in the Outflow of the upstream reservoir using the Simple Target method.

Slots Specific to this Method

☞ **Outflow from Upstream Reservoir**

Type: Series
Units: FLOW

Description: outflow from the upstream reservoir
Information:
I/O: Required input
Links: Should be linked to the Outflow slot of the upstream reservoir.

☞ TOTAL LAG OF UPSTREAM REACHES

Type: Table
Units: TIME
Description: combined lag time of reaches between the current and the upstream reservoir
Information:
I/O: Required input
Links: Not linkable

24.1.28 Sediment

The Sediment category is used to enable algorithms which adjust reservoir Elevation Volume and possibly Elevation Area relationships in response to sediment inflow. See also the “Time Varying Elevation Area” method HERE (Objects.pdf, Section 24.1.29.2) for more information on methods that change the elevation area relationship.

24.1.28.1 None

The None method is the default for the Sediment category. No calculations are performed in this method. There are no slots specifically associated with this method.

24.1.28.2 CRSS Sediment

The **CRSS Sediment** method is designed based on sedimentation calculations performed by the US Bureau of Reclamation’s Colorado River Simulation System (CRSS) model. This function distributes reservoir sediment based on the “Empirical Area Reduction Method”. Simply put, sediment is distributed through an iterative process in which a total volume loss due to sedimentation is calculated based on an assumed top of sediment elevation.

Slots Specific to this Method

☞ Elevation Area Table

Type: Table
Units: LENGTH vs. AREA
Description: generated elevation area table for calculating sediment distribution
Information:
I/O: Output only
Links: Not linkable

☞ Elevation Vol_Area Table Increment

Type: Table
Units: LENGTH
Description: elevation increments for the generated Elevation Volume and Elevation Area Tables
Information: This table often needs more precise elevation increments than the sediment calculation tables.
I/O: Required input
Links: Not linkable

☞ Initial Elevation Area Table

Type: Table

Units: LENGTH VS. AREA
Description: initial elevation area table
Information: Provided for comparison with initial data
I/O: Output only
Links: Not linkable

☞ Initial Elevation Volume Table

Type: Table
Units: LENGTH VS. VOLUME
Description: initial elevation volume table
Information: provided for comparison with initial data
I/O: Output only
Links: Not linkable

☞ Sediment Distribution Coefficients

Type: Table
Units: NOUNITS
Description: parameters for empirical equation governing sediment distribution
Information:
I/O: Required input
Links: Not linkable

☞ Sediment Inflow

Type: Series
Units: VOLUME
Description: volume of sediment flowing into the reservoir at each timestep
Information:
I/O: Required input
Links: Not linkable

☞ User Input Elev Area Data

Type: Table
Units: LENGTH VS. AREA
Description: initial Elevation Area relationship
Information: These values are initial conditions for the first timestep of the simulation. The elevation increments will be used for all sedimentation calculations.
I/O: Required input
Links: Not linkable

This volume loss is recalculated (with a new top of sediment elevation) at each iteration, until the calculated volume loss is equal to the actual volume of sediment inflow (within a specified convergence). The total volume loss calculation consists of a somewhat complicated algorithm utilizing elevation/area and elevation/volume data for the reservoir and an empirical equation. The empirical equation uses user specified parameters which relate the portion of total area that is taken up by sediment to the Pool Elevation. The empirical equation basically gives the shape of the accumulated sediment. The empirical equation has a close relationship to the elevation volume and elevation area characteristics of a given reservoir. The elevation/area and elevation/volume data is stored in a polynomial coefficient table, which gets recalculated after each timestep. The actual Elevation Area, Elevation Volume tables used by **RiverWare**™ are adjusted at the end of the sedimentation code (but prior to the hydrologic simulation).

Caution should be exercised in creating input data for this method. The close relationship between the empirical area reduction equation and the shape of the reservoir (reflected in the User Input Elev Area Data) makes the method fairly sensitive to input data. When choosing empirical parameters for this method, physical characteristics of the given reservoir need to be considered. The Bureau of Reclamation currently considers 4 possible types of reservoirs, with each type having a corresponding set of empirical area reduction parameters. The reservoir type classification is based on the shape of the Reservoir, the manner in which the reservoir is to be operated, and the size of the sediment particles to be deposited in the reservoir. The main emphasis is on the shape. Tables are used to classify the reservoirs based on these characteristics. Once the type has been established, the parameter values for that type can also be taken from tables in the literature. An incorrect set of parameters for a given reservoir will lead to an inability to achieve convergence on the sediment distribution within this method.

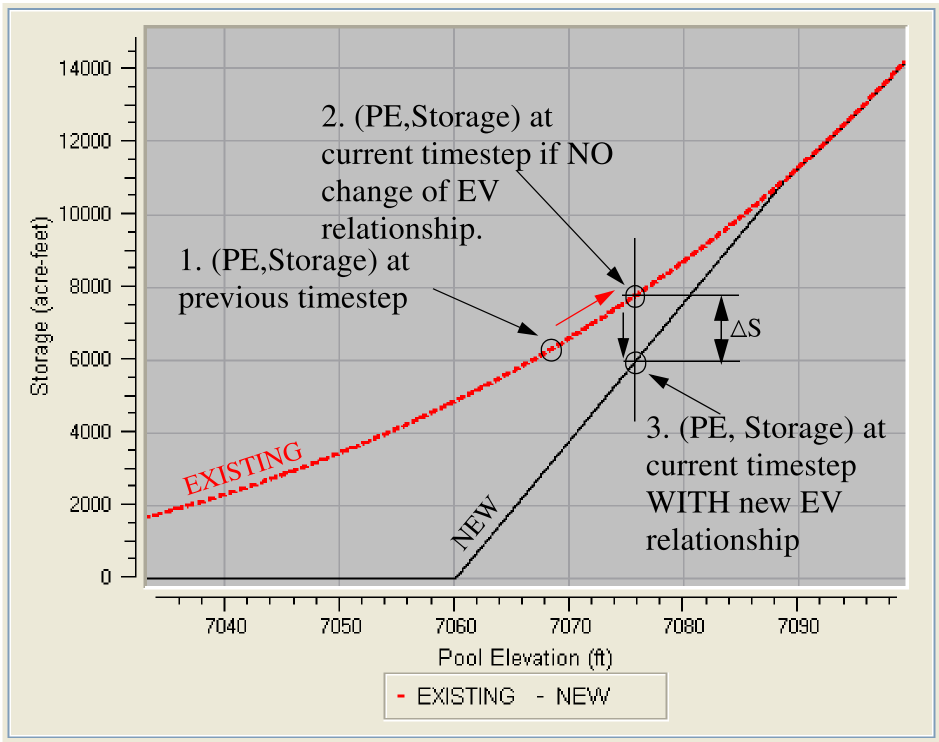
24.1.28.3 Time Varying Elevation Volume

The **Time Varying Elevation Volume** method allows Elevation Volume (EV) relationships that change at specified times. The method is only available when the following default methods are selected in the following categories:

- Flood Control Release: only the None method is allowed
- Surcharge Release: only the None method is allowed
- Water Quality: None, Water Quality cannot be enabled

In the figure below, both the existing and new EV relationships are plotted. Shown is a graphical example of how the mass balance should be performed on the timestep the EV relationship changes. This process is described below.

1. This is the PE and Storage at the previous timestep. This is the starting condition.



2. On the current timestep, the Inflows and Outflows lead to a positive net inflow. This will increase the Pool Elevation. Using the existing table, a (PE, Storage) is calculated and shown as point 2.

3. The PE is used to interpolate a storage on the new EV table. The difference between storage at point 2 and point 3 is the loss of storage term. This means that the Pool Elevation is the same regardless of which table is used.

Slots Associated With This Method

☞ Elevation Volume Table Time Varying

Type: Table Slot

Units: Length and volume

Description: the tables that represent the Elevation Volume relationship at various times in the run.

Information: The number of columns in this table should be set to one plus the number of times the Elevation Volume relationship changes. The column headings contain the date corresponding to the change. When you add a column to this slot, it is given a date later then the last column. You can set the date from the Column -> Set Column Value menu. You can then type the date text or use the date time spinner to enter the appropriate date. The dialog will only let you enter dates that correspond to timesteps. (Care should be exercised when switching model timesteps) The column is then placed in the correct order compared to the other column labels. Each column should have an entry for each row in the Elevation column or an interpolation error may be issued during the run. The number of columns is equal to the number of changes plus one, i.e., if there is one change then there should be two columns. An example is shown below for a run that starts in 24:00 Jan 1, 1910 and the reservoir Elevation Volume changes three times. The column label of the first set of volumes must be on or before the initial timestep.

I/O: Input Only

Links: Not Linkable

Pool Elevation ft	24:00 Jan 1, 1910 Storage acre-ft	24:00 Jan 1, 1935 Storage acre-ft	24:00 Jan 1, 1953 Storage acre-ft	24:00 Jan 1, 1970 Storage acre-ft
5,100	0	0	0	0
5,120	10	9	8	7
5,150	15	14	13	12
5,200	20	19	18	17
5,250	25	24	23	22

☞ Storage Adjustment from Elev Vol Table Change

Type: Series Slot

Units: VOLUME

Description: This is the volume of water that was lost to sedimentation

Information: The slot tracks the mass discontinuity that occurs when the Elevation Volume is changed because of a new reservoir Elevation Volume relationship. A positive number indicates storage was gained; a negative number indicates storage was lost.

I/O: Output Only

Links: Not Linkable

Method Details

Start of Run: The run proceeds as follows: at the start of the run, if the **Time Varying Elevation Volume** method is selected, a boolean variable isTimeVaryingElevVolume will be set to TRUE. At the same time, a pointer is set that specifies that all computations should use the **Elevation Volume Table Time Varying**. All computations involving the elevation volume relationship on the object use this pointer instead of the directly accessing the **Elevation Volume Table**. If the method is not selected, then the pointer is set to the **Elevation Volume Table**.

> **Note:** Even with this method selected, the original Elevation Volume Table is still visible. Although it is not used when dispatching or other simulation or rulebased simulation, it is still a general slot that is used in optimization calculations and water quality.

Start of Each Dispatch Method: At the start of each dispatch method, if the isTimeVaryingElevVolume is true, then the method will determine which column ,"col", of the table to use. It compares the current timestep to the column headings on the table and determine which column to use. For example, if the current timestep is March 3, 1940 for the table example above, then it will set col = 2 (column numbering is zero based) for use in all remaining computations. If the current timestep exactly matches one of the column headings, then an additional variable, isElevVolModDate, is set to TRUE and col is set to the that column minus 1. That is, for this dispatch, the previous relationship will be used, but will be adjusted at the end of the method. Note, if the **Time Varying Elevation Volume** method is not selected, then isTimeVaryingElevVolume and isElevVolModDate will remain false and the column to use on the **Elevation Volume Table** is set to 1.

The dispatch method then proceeds as normal using the computed "col" in all references to the specified elevation volume table.

End of Each Dispatch Method: At the end of the dispatch method (the description applies first to solveMB_GivenInflowOutflow), once PE and Storage are known, if the isElevVolModDate is true, the method will lookup the PE on the **Elevation Volume Table Time Varying** but this time use col+1 and get S'. The new storage S' is the reduced storage using the new EV relationship. The difference, $\Delta S = S - S'$, and is set on the **Storage Adjustment from Elev Vol Table Change** slot. Then the Storage slot is set to S' and PreviousStorage[t+1] is set to S'.

The above procedure describes the solveMB_givenInflowOutflow. This same procedure is used for: solveMB_givenEnergyInflow and solveMB_givenInflowRelease. Once the new storage using the existing table is calculated, the new relationship can be used. If Pool Elevation (HW) is given (i.e. solveMB_givenOutflowHW), then the Pool Elevation is used to look up the storage using the existing and new tables. For the dispatch methods where Storage is known, the method will abort the run with an error that the **Time Varying Elevation Volume** method cannot be used on a timestep where the storage is given.

Limitations: This method changes fundamental information about the reservoir. As a result, there are certain operations that cannot be used with this method including:

- Target operations that span table modification dates
- Dispatching the reservoir with any of the "givenStorage" methods on a modification date. Non modification dates can use the "givenStorage" methods.
- Any of the following RPL functions: StorageToElevation, ElevationToStorage, and StorageToArea. If these functions are called on a reservoir with the **Time Varying**

Elevation Volume selected, an error message will be posted. Instead use the "...AtDate" version of that function. I.e. use the StorageToElevationAtDate instead of the StorageToElevation function. Old models may need to be updated.

- There are many RPL functions like SolveOutflow, SolveStorage, GetMaxOutflowGivenInflow, etc that access the elevation volume relationship. These will access the correct table, but will always assume that the computation is being performed BEFORE any modifications to the relationship are made. That is, if you call the function and it is a modification timestep on the table, the function will use the previous column in all its computations. The relationship change is only considered at the end of the dispatch method, not in the RPL function.

24.1.29 Surface Area Modification

The **Surface Area Modification** category is dependent on having a valid evaporation method selected (i.e. any evaporation method except **None** or **MonthlyEvaporationCalcInAnnual**). The category will be added to all reservoirs and the default method in this category is **None** which performs no calculations and has no slots.

24.1.29.1 None

This is the default, no-action method.

24.1.29.2 Time Varying Elevation Area

The **Time Varying Elevation Area** method allows Elevation Area (EA) relationships that change at specified times. The **Time Varying Elevation Area** method is only available on Level Power and Storage reservoirs. Also, the method will only be available when the following default methods are selected in the following categories:

- Flood Control Release: only the None method is allowed
- Surcharge Release: only the None method is allowed
- Water Quality: None; Water Quality must be disabled

Slots Associated with this Method

☞ **Elevation Area Table Time Varying**

Type: Table Slot

Units: Length and Area

Description: the tables that represent the Elevation Area relationship at various times in the run.

Information: The number of columns in this table should be set to one plus the number of times the Elevation Area relationship changes. The column headings contain the date corresponding to the change. When you add a column to this slot, it is given a date later then the last column. You can set the date from the Column -> Set Column Value menu. You can then type the date text or use the date time spinner to enter the appropriate date. The dialog will only let you enter dates that correspond to timesteps. (Care should be exercised when switching model timesteps.) The column is then placed in the correct order compared to the other column labels. Each column should have an entry for each row in the Pool Elevation column or an interpolation error may be issued during the run. The number of columns is equal to the number of changes plus one, i.e., if there is one change then there should be two columns. An example is shown below for a run that starts in Jan 1, 1910 and the reservoir Elevation Area changes three times. The column label of the first column of areas must be on or before the initial timestep. The times on the column map are an instant in time.

I/O: Input Only

Links: Not Linkable

Pool Elevation ft	24:00 Jan 1, 1910 Surface Area acre	24:00 Jan 1, 1935 Surface Area acre	24:00 Jan 1, 1953 Surface Area acre	24:00 Jan 1, 1970 Surface Area acre
5,100	0	0	0	0
5,120	10	9	8	7
5,150	15	14	13	12
5,200	20	19	18	17
5,250	25	24	23	22

☞ Surface Area Adjustment to Elev Area Table Change

Type: Series Slot

Units: AREA

Description: The surface area lost to sedimentation that occurs when the reservoir is resurveyed and a new Elevation Area Table is implemented.

Information: A positive number indicates area was added. A negative number indicates surface area was lost.

I/O: Output Only

Links: Not Linkable

Method Details

The evaporation methods typically use the average Surface Area in the calculations. Because the average Surface Area may change when the new table becomes active, the computations assume that the Surface Area only changes at the end of a dispatch method. That is, on a modification timestep, the previous relationship is used in the evaporation and precipitation methods, then the Surface Area is modified at the end of the dispatch method. Thus, the modified surface area will be reflected in evap and precip computations at the NEXT timestep.

The run proceeds as follows:

Start of Run: At the start of the run, if the **Time Varying Elevation Area** method is selected, a boolean variable isTimeVaryingElevArea will be set to TRUE. At the same time, a pointer is set that specifies to use the **Elevation Area Table Time Varying** table. All other computations now reference this pointer instead of the **Elevation Area Table**. If the method is not selected, then the pointer is set to the **Elevation Area Table**.

Note: Even with this method selected, the original Elevation Area Table is still visible. Although it is not used when dispatching or other simulation or rulebased simulation, it is still a general slot that is used in optimization calculations and water quality.

In each Evap/Precip Method: At the start of each method, if the isTimeVaryingElevArea is true, then the method will determine which column ,“col”, of the table to use. It will compare the current timestep to the column headings on the table and determine which column to use. For example, if the current timestep is March 3, 1940 for the table example above, then it will set col = 2 (column numbering is zero based) for use in all remaining computations.

If the current timestep exactly matches one of the column headings, col is set to that column minus 1. That is, for this method, the previous relationship will be used. Also, the variable isElevAreaModDate, is set to TRUE. The computation proceeds as normal. That is, the estimate for surface area uses the original relationship, not the modified relationship.

End of Dispatch method: At the end of the dispatch method, if the isElevAreaModDate is true, the method will lookup the PE on the **Elevation Area Table Time Varying** but this time use col+1 and get SA’. The new surface area SA’ is the modified surface area using the new EA relationship. The difference, ΔSA = SA - SA’, is set on the **Surface Area Adjustment to Elev Area Table Change** slot. Then the Surface Area slot is set to SA’. This does not affect the evaporation or precipitation computation on this timestep.The new relationship will be used on the next timestep though.

Note, if the **Time Varying Elevation Area** is not selected, then isTimeVaryingElevArea and isElevAreaModDate will remain false and the column to use on the **Elevation Area Table** is set to 1.

Limitations: This method changes fundamental information about the reservoir. As a result, there are certain operations that cannot be used with this method including:

- Any of the following RPL functions: StorageToArea and ElevationToArea. If these functions are called on a reservoir with the **Time Varying Elevation Area** method selected, an error message will be posted. Instead use the “...AtDate” version of that function. I.e. use the ElevationToAreaAtDate instead of the ElevationToArea function.
- There are many RPL functions like SolveOutflow, SolveStorage, GetMaxOutflowGivenInflow, etc that access the elevation area relationship. These will access the correct table, but will always assume that the computation is being performed BEFORE any modifications to the relationship are made. That is, if you call the function and it is a modification timestep on the table, the function will use the previous column in all its computations. The relationship change is only considered at the end of the dispatch method, not in the RPL function. The new relationship will be used on the next timestep though.

24.1.30 Disable Reservoir Processes

This category holds the Pass Inflows method which allows you to disable reservoir processes and pass inflows as though the reservoir wasn't there.

24.1.30.1 None

This is the default, no-action method. When this method is selected all physical processes are enabled and executed as appropriate.

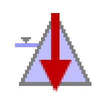
24.1.30.2 Pass Inflows

This method disables many of the reservoir physical processes and passes the inflows through the reservoir. This method can be used to temporarily remove the presence of a reservoir and compute flows as though they were unregulated, unconstrained or 'pre-project'.

No Slots are associated with this method.

This method does the following:

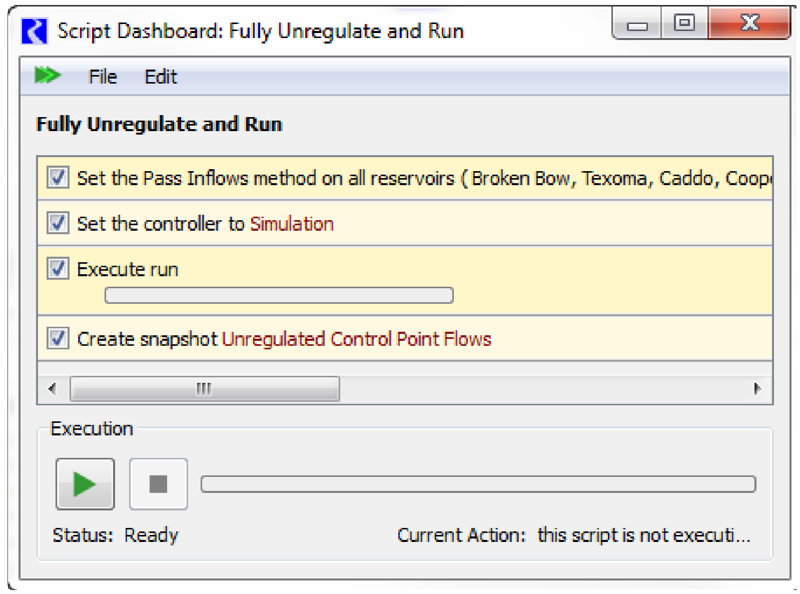
- **Changes the Workspace Icon**: When this method is selected, the icon on the workspace is modified to indicate the Reservoir is passing inflows.



- **Enables alternative dispatch method**: When this method is selected, ONLY the **Outflow Equals Sum of Inflows** dispatch method is available. This dispatch method is described HERE (Section 24.2.9).

Computation of flows for a Fully Unregulated system. To compute the flows at any point in the system without the effect of any reservoir, run two simulations. The first run computes the unregulated flows, the second computes the regulated flows, often using rulebased simulation. Results can be analyzed using snapshots. Simple scripts can be used to modify the system and make the runs. Following is the conceptual approach:

1. Compute unregulated flows by executing a script. Scripts are described HERE (ScriptManagement.pdf, Section 1). A screenshot of a sample **Script Dashboard** is shown. It does the following:



- Set the Pass Inflows method on each reservoir.
- Set the run controller to **Simulation**
- Run the model. With the new method, the reservoirs pass inflows (including Hydrologic Inflow and Return Flow) but no other physical processes are made. When the reservoirs dispatch, they will set Outflow = ΣInflows, thus propagating the flows downstream.
- Create snapshots of desired slots.

2. You could then look at the results if desired.

3. Compute regulated flows by executing a script which does the following:

- Set the method on each reservoir to its normal regulated mode (**None** method in the **Disable Reservoir Processes** category).
- Set the run control to **Rulebased Simulation**
- Load the RBS ruleset set if necessary
- Run the model

4. At this point, the regulated results are in the objects and slots. They can be compared with the snapshots from the unregulated run.

The process above assumes two scripts, steps 1 and 3; these could be combined into one script that fully automates the runs if desired. With this approach it is easy for the user to run just the regulated system, the unregulated system or both.

This approach presented assumes that all reservoirs are unregulated. If only a portion of the system is unregulated, then you will need to decide how the other reservoirs behave, especially if they are dependent on the disabled reservoir. Additional script actions can be used to disable rules or set values in the system. The USACE approach to unregulated conditions is described HERE (USACE_SWD.pdf, Section 4).

24.2 Dispatch Methods

24.2.1 solveMB_givenInflowHW

This dispatch method is placed on the queue when the list of knowns/unknowns (below) is met. It solves for Outflow.

Required Knowns

- **Diversion**
- **Return Flow**
- **Inflow**
- **Pool Elevation**
- **Hydrologic Inflow**

(only if input or Solve Hydrologic Inflow is selected)

Required Unknowns

- **Storage**
- **Outflow**

First, this method finds the Storage associated with the known Pool Elevation using the Elevation Volume Table. It then checks if the Pool Elevation at the current timestep is flagged as a TARGET. If it is, the target calculations are performed. If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated using the other parameter. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. If a Canal Object is linked and not yet solved, the dispatch method is exited and waits for the Canal to solve.

The mass balance is carried out by the following equation:

$$Outflow = Storage(-1) - Storage + Inflow$$

This equation may or may not contain other elements reflecting the user method choices (including Hydrologic Inflow, Evaporation, Precipitation, Seepage, and Bankstorage). Inflow in the mass balance equation is the net inflow to the reservoir, including hydrologic inflows, return inflows, and diversions if appropriate. Spill and Release are then found according to the user selected methods.

The Energy in Storage, Spilled Energy, and Future Value calculation user methods are then executed.

24.2.2 solveMB_givenOutflowHW

This dispatch method is placed on the queue when the list of knowns/unknowns (below) is met. It solves for Inflow.

Required Knowns

- ☞ **Diversion**
- ☞ **Return Flow**
- ☞ **Outflow**
- ☞ **Pool Elevation**
- ☞ **Hydrologic Inflow**

(only if input or Solve Hydrologic Inflow is selected)

Required Unknowns

- ☞ **Storage**
- ☞ **Inflow**

First, this method finds the Storage associated with the known Pool Elevation using the Elevation Volume Table. If the Pool Elevation at the current timestep is flagged as a TARGET, the target calculations are performed. It either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated using the other parameter. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. If a Canal Object is linked and has not yet solved, the dispatch method is exited and waits for the Canal to solve.

The Outflow is then checked for the Max Capacity flag. If this flag is present, the maximum outflow is found from the given Pool Elevation. Outflow is calculated using the following equation:

$$Outflow = maximum\ Spill + maximum\ Turbine\ Release$$

Then the mass balance is carried out using the following equation:

$$Inflow = Storage - Storage(-1) + Outflow$$

This equation may or may not contain other elements reflecting the user method choices (including Hydrologic Inflow, Evaporation, Precipitation, Seepage, and Bankstorage). Inflow in the mass balance is the net inflow to the reservoir, including the hydrologic inflows, return flows, and diversions if appropriate. The Inflow slot is adjusted accordingly to account for these terms. Spill and Release are then found according to the user selected methods.

If the value found for the Inflow is less than zero, the hydrologic inflow calc method is checked. If it is Hydrologic Inflow and Loss, the method is called, setting the Inflow to zero, and the Hydrologic Inflow Adjust to the previous value of the inflow. This is done to present the negative inflow as a loss that has not been accounted for.

The Energy in Storage, Spilled Energy, and Future Value calculation user methods are then executed.

24.2.3 solveMB_givenInflowStorage

This dispatch method is placed on the queue when the list of knowns/unknowns (below) is met. It solves for Outflow.

Required Knowns

- **Diversion**
- **Return Flow**
- **Inflow**
- **Storage**
- **Hydrologic Inflow**

  (only if input or Solve Hydrologic Inflow is selected)

Required Unknowns

☞ **POOL ELEVATION**

☞ **OUTFLOW**

First, this method finds the Pool Elevation associated with the known Storage using the Elevation Volume Table. If the Storage at the current timestep is flagged as a TARGET, the target calculations are performed. If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated using the other parameter. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. If a Canal Object is linked and has not yet solved, the dispatch method is exited and waits for the Canal to solve.

The mass balance is carried out using the following equation:

$$Outflow = Storage(-1) - Storage + Inflow$$

This equation may or may not contain other elements reflecting the user method choices (including Hydrologic Inflow, Evaporation, Precipitation, Seepage, and Bankstorage). Inflow in the mass balance equation is the net inflow to the reservoir, including hydrologic inflows, return flows, and diversions if appropriate. If the Outflow is less than zero, the method is exited. Spill and Release are then found according to the user selected methods.

The Energy in Storage, Spilled Energy, and Future Value calculation user methods then executed.

This method will issue an error if the "Time Varying Elevation Volume" method is selected, HERE (Objects.pdf, Section 24.1.28.3), and the timestep is a modification date on the table.

24.2.4 solveMB_givenOutflowStorage

This dispatch method is placed on the queue when the list of knowns/unknowns (below) is met. It solves for Inflow.

REQUIRED KNOWNS

☞ **Diversion**

☞ **Return Flow**

☞ **Storage**

☞ **Outflow**

☞ **Hydrologic Inflow**

(only if input or Solve Hydrologic Inflow is selected)

Required Unknowns

☞ **Pool Elevation**

☞ **Inflow**

First, this method finds the Pool Elevation associated with the known Storage using the Elevation Volume Table. If the Storage is flagged as a TARGET and Outflow is not flagged as MAX CAPACITY, the target calculations are performed. If the Storage is flagged as a TARGET and the Outflow is flagged MAX CAPACITY, a **RiverWare**™ error is posted and the run is aborted because an under-determination would result. If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated using the other parameter. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. If a Canal Object is linked and has not yet solved, the dispatch method is exited and waits for the Canal to solve.

The outflow is checked for the Max Capacity flag. If this flag is present, the maximum outflow is found from the given Pool Elevation. Outflow is calculated using the following equation:

$$Outflow = maximum\ Spill + maximum\ Turbine\ TurbineRelease$$

Then the mass balance is carried out:

$$Inflow = Storage - Storage(-1) + Outflow$$

This equation may or may not contain other elements reflecting the user method choices (including Hydrologic Inflow, Evaporation, Precipitation, Seepage, and Bankstorage). Inflow in the mass balance equation is the net inflow to the reservoir, including hydrologic inflows, return flows, and diversions if appropriate. The Inflow slot is adjusted accordingly to account for these additional terms. Spill and Release are then found according to the user selected methods.

If the value found for the inflow is less than zero, the hydrologic inflow calc method is checked. If it is Hydrologic Inflow and Loss, the method is called, setting the Inflow to zero, and the Hydrologic Inflow Adjust to the previous value of the Inflow. This is done to present the negative inflow as a loss that has not been accounted for.

The Energy in Storage, Spilled Energy, and Future Value calculation user methods are then executed.

This method will issue an error if the "Time Varying Elevation Volume" method is selected, HERE (Objects.pdf, Section 24.1.28.3), and the timestep is a modification date on the table.

24.2.5 solveMB_givenInflowOutflow

This dispatch method is placed on the queue when the list of knowns/unknowns (below) is met. It solves for Storage (and Pool Elevation).

Required Knowns

- **Diversion**
- **Return Flow**
- **Inflow**
- **Outflow**
- **Hydrologic Inflow**

(only if input or Solve Hydrologic Inflow is selected)

Required Unknowns

- **Pool Elevation**
- **Storage**

First, this method checks if there are Pool Elevation and Storage values for the previous timestep. If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated using the other parameter. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. If a Canal Object is linked and has not yet solved, the dispatch method is exited and waits for the Canal to solve.

The Outflow is then checked for the Max Capacity flag. If this flag is present, the maximum outflow is found from the mass balance and spill methods using a convergence algorithm. The algorithm used is detailed **HERE (Appendix A: Reservoir Convergence)**. The Outflow slot is set equal to the maximum outflow value.

If the Outflow slot is set by the Surcharge Release flag, the surcharge release is computed and the Outflow slot is set equal to surcharge release for all timesteps in the forecast period (the surcharge release methods compute a surcharge release forecast). Additional information on dispatching when using the Surcharge Release Flag can be found **HERE (USACE_SWD.pdf, Section 3.3.3)**.

Then the mass balance is carried out:

$$Storage = Storage(-1) + Inflow - Outflow$$

This equation may or may not contain other elements reflecting the user method choices (including Hydrologic Inflow, Evaporation, Precipitation, Seepage, and Bankstorage). Spill and Release are then found according to the user selected methods.

The Energy in Storage, Spilled Energy, and Future Value calculation user methods are then executed.

24.2.6 solveMB_givenInflowRelease

This dispatch method is placed on the queue when the list of knowns/unknowns (below) is met. It solves for Storage (and Pool Elevation) and Outflow.

Required Knowns

☞ **Diversion**

☞ **Return Flow**

☞ **Inflow**

☞ **Total Release**

☞ **Hydrologic Inflow**

(only if input or Solve Hydrologic Inflow is selected)

Required Unknowns

☞ **Pool Elevation**

☞ **Storage**

☞ **Outflow**

First, this method checks if there are Pool Elevation and Storage values for the previous timestep. If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated using the other parameter. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. If a Canal Object is linked, the method exits with an error, because this dispatch method is not valid when a Canal is linked to the reservoir.

If there is an unregulated spillway crest (because an unregulated spill method is selected) the method computes the upper limit for unregulated spill to prevent spilling too much water to drop the reservoir below the crest. Typically, this limit is computed as the volume of water above the crest converted to a flow. See the Unregulated Spill method for details.

At this point, the method enters a convergence algorithm which iterates to find an outflow and storage that satisfies both the mass balance equation and the release/spill curve. The convergence algorithm is detailed HERE (Appendix A: Reservoir Convergence).

The Energy in Storage, Spilled Energy, and Future Value calculation user methods are then executed.

24.2.7 solveMB_givenInflowOutflowStorage

This dispatch method is only available when the Solve Hydrologic Inflow method is selected. It is placed on the queue when the list of knowns/unknowns (below) is met. It solves for the Hydrologic Inflow.

Required Knowns

☞ **Diversion**

☞ **Return Flow**

☞ **Outflow**

☞ **Storage**

☞ **Inflow**

Required Unknowns

☞ **Pool Elevation**

☞ **Hydrologic Inflow**

First, this method checks if Outflow is flagged as MAX CAPACITY. If so, a **RiverWare**™ error is flagged and the run is aborted. Then, the Pool Elevation associated with the known Storage is determined from the Elevation Volume Table. If the Storage is flagged as TARGET, a **RiverWare**™ error is posted and the run is aborted. If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated using the other parameter. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. If a Canal Object is linked, a **RiverWare**™ error is posted and the run is aborted.

The the hydrologic inflow is found with the following equation (all values are converted to flows):

$$HydrologicInflow = Storage - Storage(-1) + Outflow - Inflow$$

This equation may or may not contain other elements reflecting the user method choices (these include Evaporation (+), Precipitation(-), Seepage(+), Bankstorage(+), Return Flow(-), Diversion (+), and Flow TO/FROM Pumped Storage). Spill and release are then found according to the user selected methods.

The Energy in Storage, Spilled Energy, and Future Value calculation user methods are then executed.

This method will issue an error if the "Time Varying Elevation Volume" method is selected, HERE (Objects.pdf, Section 24.1.28.3), and the timestep is a modification date on the table.

24.2.8 solveMB_givenInflowOutflowHW

This dispatch method is only available when the Solve Hydrologic Inflow method is selected. It is placed on the queue when the list of knowns/unknowns (below) is met. It solves for the Hydrologic Inflow.

REQUIRED KNOWNS

- **DIVERSION**
- **RETURN FLOW**
- **OUTFLOW**
- **POOL ELEVATION**
- **INFLOW**

REQUIRED UNKNOWNS

- **STORAGE**
- **HYDROLOGIC INFLOW**

First, this method checks if Outflow is flagged as MAX CAPACITY. If so, a **RiverWare**™ error is flagged and the run is aborted. Then, the Storage associated with the known Pool Elevation is determined from the Elevation Volume Table. If the Pool Elevation is flagged as TARGET, a **RiverWare**™ error is posted and the run is aborted. If either the previous timestep's Pool Elevation or the previous timestep's Storage is not known, it is calculated using the other parameter. If both the previous timestep's Pool Elevation and the previous timestep's Storage are unknown, the dispatch method is exited. If a Canal Object is linked, a **RiverWare**™ error is posted and the run is aborted.

The the hydrologic inflow is found with the following equation (all values are converted to flows):

$$HydrologicInflow = Storage - Storage(-1) + Outflow - Inflow$$

This equation may or may not contain other elements reflecting the user method choices (these include Evaporation (+), Precipitation(-), Seepage(+), Bankstorage(+), Return Flow(-), Diversion (+), and Flow TO/FROM Pumped Storage).

Spill and release are then found according to the user selected methods and the equation:

$$Outflow = Spill + Release$$

The energy in storage, spilled energy, and future value calculation user methods are called. The dispatch method is then complete.

24.2.9 Outflow Equals Sum of Inflows

This dispatch method is only available when the **Pass Inflows** method HERE (Section 24.1.30.2) is selected. It has the following dispatch conditions:

Required Knowns

☞ **Inflow**

Required Unknowns

☞ **Outflow**

This method does the following:

An error is issued if any of the following are linked, specified as inputs, and/or are not zero:

- Canal Flow,
- Diversion,
- Flow TO Pumped Storage,
- Flow FROM Pumped Storage
- Seepage

An error is issued if:

- Any flags are set on Outflow (an over-determination error will be issued).
- Water Quality is enabled on this reservoir

Inflow Sum and Total Inflows are computed as usual as described HERE (Total Inflows)

Finally:

$$\begin{aligned} \text{Outflow} = {} & \text{Inflow} + \text{Return Flow} + \\ & \text{Hydrologic Inflow} + \text{Hydrologic Inflow Adjust} + \text{Hydrologic Inflow Forecast} \end{aligned}$$

Then the dispatch method finishes successfully and exits.

Note: Hydrologic Inflow disaggregation described HERE (Section 24.1.11) and forecasting described HERE (Section 24.1.10) is performed at beginning of the run or the timestep. Therefore, these inflows to the reservoir are included in the outflow.

But, none of the other physical process methods are executed even though they may have method selections and slots visible. These include:

- Precipitation and Evaporation
- Spill
- Energy in Storage
- Seepage
- Bank Storage
- Tailwater
- Operating Level

25. Stream Gage

This object models a measuring point in a water system. The **Gage Inflow** is automatically linked to the **Gage Outflow**.

General Slots

Slots which always appear for this object:

☞ Gage Inflow

Type: SeriesSlot

Units: FLOW

Description: flow into the gage

Information:

I/O: Optional; either Inflow or Outflow must be known.

Links: May be linked to the Outflow of any object.

☞ Gage Outflow

Type: AggSeriesSlot

Units: FLOW

Description: flow out of the gage

Information:

I/O: Optional; either Inflow or Outflow must be known.

Links: May be linked to the Inflow of any object

If both **Gage Outflow** and **Gage Inflow** are input, a **RiverWare** error is posted and the run is aborted.

25.1 User Methods

25.1.1 Conditional Flow

The Conditional Flow category allows the **Gage Inflow** slot to be set if the **Fractional Flow** method has been selected.

25.1.1.1 None

The **None** method is the default for the category. This method does not add or set any slots.

25.1.1.2 Fractional Flow

This method, called from the **Solve Gage Flow** dispatch method, compares two user specified flow conditions and sets the **Gage Inflow** slot. Once the **Gage Inflow** slot has been set, that value will immediately propagate to the **Gage Outflow** slot because the two are linked.

Slots Specific To this Method

☞ **Condition One**

Type: Series Slot

Units: FLOW

Description: A slot for comparing two conditions that will affect the Gage Inflow slot

Information: This is a dispatch slot that will trigger the object to resolve if its value changes. Values are received via: links, user input, or set by rules.

I/O: Required Input

Links: linkable

☞ **Condition Two**

Type: Series Slot

Units: FLOW

Description: A slot for comparing two conditions that will affect the Gage Inflow slot

Information: This is a dispatch slot that will trigger the object to resolve if its value changes. Values are received via: links, user input, or set by rules.

I/O: Required Input

Links: linkable

☞ **Normal Flow**

Type: Series Slot

Units: FLOW

Description: Timeseries of values of Inflow.

Information: When slot Condition One is equal to or greater than slot Condition Two, the Gage Inflow is set to Normal Flow.

I/O: Required Input

Links: linkable

☞ Loss Factor

Type: Periodic Slot

Units: NONE

Description: schedule of factors specifying fraction of water remaining after conveyence losses upstream of the gage object. Different rows represent different time points in the schedule, while one column specifies **Fraction** and the other column specifies **Constant**.

Information: This is a n x 2 periodic slot where the user specifies the period over which the schedule repeats(yearly, monthly, etc.) and the slot has the capability to generate values for any date from the schedule.

I/O: Required Input

Links: not linkable

In the **Fractional Flow** method, **Gage Inflow** is conditional, determined through a comparison between the values in the slots **Condition One** and **Condition Two**. If the value in the **Condition Two** slot is greater than the value in the **Condition One** slot, the **Gage Inflow** slot is set to **Constant** plus **Condition One** multiplied by **Fraction**. Otherwise the **Gage Inflow** slot is set to value in the **Normal Flow** slot.

The following displays how the **Gage Inflow** is calculated:

IF (**Condition One** < **Condition Two**)

$$\text{Gage Inflow} = Constant + (ConditionOne \times Fraction)$$

ELSE

$$\text{Gage Inflow} = \text{Normal Flow}$$

25.1.2 Stream Gage Stage

The **Stream Gage Stage** category contains methods allowing you to configure optional stage computations. The methods in the category are executed by both dispatch methods.

25.1.2.1 None

The **None** method is the default for the category. This method does not add or set any slots. No stage is computed.

25.1.2.2 Stage Table Lookup

The new **Stage Table Lookup** method looks up the **Gage Inflow** on a table and then sets the **Stage** slot.

SLOTS ASSOCIATED WITH THIS METHOD:

☞ **STAGE TABLE**

Type: Table
Units: FLOW VS LENGTH
Description: This table represents the relationship between flow and gage height.
Information:
I/O: Required Input
Links: Not Linkable

☞ **STAGE**

Type: Series
Units: LENGTH
Description: The timeseries of the stream gage water surface elevation
Information: The elevation datum is up to the user. If you are comparing to other objects, then a consistent datum is necessary.
I/O: Output only
Links: Linkable

METHOD DETAILS:

When executed, this method looks up the **Gage Inflow** on the **Stage Table** and sets the **Stage** slot.

Note: This method only computes the Stage, it does not do the reverse lookup to compute the flow from the Stage.

25.2 Dispatch Methods

The dispatch methods are available if either of the two user categories has a non-default method selected (i.e. any method selected except **None**)

25.2.1 Solve Gage Flow

This method is available when the **Fractional Flow** method is selected. This method solves for **Gage Inflow** when **Condition One** and **Condition Two** of the Fractional Flow method are known. The value set in the **Gage Inflow** slot automatically propagates to the **Gage Outflow** slot.

REQUIRED KNOWNS:

- **CONDITION ONE**
- **CONDITION TWO**

REQUIRED UNKNOWNS:

- **INFLOW**

METHOD DETAILS:

The dispatch method will simply execute the selected **Conditional Flow** method and set **Gage Inflow** if it has been computed by the **Conditional Flow** method.

It then calls the selected **Stream Gage Stage** method. If the **Stage Table Lookup** method is selected, the **Stage** is set.

25.2.2 Solve Gage

This method is only available when the **Stage Table Lookup** method is selected, but the **Fractional Flow** method is not selected. Since **Gage Inflow** is linked to **Gage Outflow,** the only purpose of this dispatch method is to execute the selected **Stage** method.

REQUIRED KNOWNS:

- **GAGE INFLOW**

REQUIRED UNKNOWNS: NONE

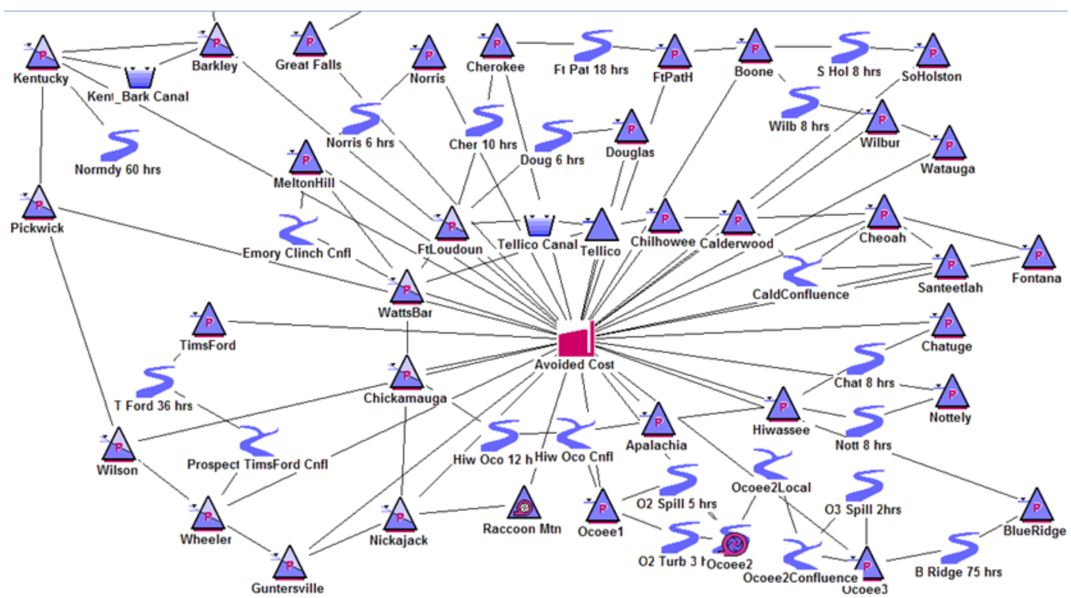
METHOD DETAILS:

The dispatch method calls the selected **Stream Gage Stage** method. If the **Stage Table Lookup** method is selected, the **Stage** is set.

26. Thermal

This object models the economics of power generation. One methods evaluates hydro generation by directly replacing power generated by thermal power plants such as nuclear, coal, or gas powered plants and examines the economic savings at the thermal plants. This is the origin of the “thermal” name for this object. Another method place a piecewise linear or “block” value on hydropower generation. Typically, the user calculates these block values based on replaced thermal generation or power purchase opportunities, but the source doesn’t matter for the RiverWare model. A third method places a linear value on hydropower generation.

The thermal object is intended to be a singleton (i.e. one thermal object per model), including all of the relevant economic information and linked to all of the power reservoirs. The thermal object provides an economic evaluation for the entire RiverWare model. In addition to economic evaluation, the thermal object is a useful place to summarize the power data of a model. The thermal object has many power related multislots and each multislot can be linked to the individual power related slots on the power reservoirs in a model. The figure below shows a red thermal object linked to power reservoirs. Like all links, these links may be hidden, dotted, or displayed in a different way to change the relative emphasis of the links in the model. Click HERE (Workspace.pdf, Section 7) for more information on link groups.



These linked multislots then provide the system totals for these power attributes. Some of the power attributes that may be linked include hydro generation, hydro capacity, pumping generation pumping capacity, and energy in storage. See the full list of multislots below for more detail.

A thermal object may be used with any of the controllers: Simulation, Rulebased Simulation, or RPL-based Optimization.

In a simulation or rule-based simulation model, a thermal object provides an economic evaluation of a solution, but does not influence the solution. An important difference with many other object classes is that the only calculations performed during the timesteps of a run are the multislot totals. All of the economic calculations are performed at the end of the run, after simulation has set all of the other slot values. One consequence of this is that the economic calculations cannot be used to drive a simulation model.

The thermal object is used in most optimization models, because economics is usually a factor in these models. Unlike simulation, in optimization the thermal object can influence the solution. In addition to allowing a modeler to maximize the economic value of hydropower, the thermal object allows a modeler to write constraints and objectives on the power related multislots. For example, total system hydropower generation or capacity can be maximized or constrained to meet a minimum level.

General Slots

(slots which always appear for this object)

☞ Energy In Storage

Type: MultiSlot
Units: ENERGY
Description: The sum of the energy stored in the linked reservoirs.
Information:
I/O: Output
Links: Linked to the Energy In Storage slot on power reservoirs.

☞ Hydro Capacity

Type: MultiSlot
Units: POWER
Description: The sum of the power generated by the linked reservoirs.
Information:
I/O: Output
Links: Linked to the Power slot on power reservoirs.

☞ Hydro Generation

Type: MultiSlot
Units: ENERGY
Description: The sum of the energy generated by the linked reservoirs.
Information:
I/O: Output

Links: Linked to the Energy slot on power reservoirs.

☞ Objective Values

Type: Table Slot
Units: NONE
Description: The optimal objective function values from the last optimization run.
Information:
I/O: Output
Links: Not linkable.

☞ Pumped Storage Generation

Type: MultiSlot
Units: ENERGY
Description: The sum of the energy generated by the linked pumped storage reservoirs.
Information:
I/O: Output
Links: Linked to the Energy slot on pumped storage reservoirs.

☞ Pumped Storage Generation Capacity

Type: MultiSlot
Units: POWER
Description: The sum of the power generated by the linked pumped storage reservoirs.
Information:
I/O: Output
Links: Linked to the Power slot on pumped storage reservoirs.

☞ Pumped Storage Pumping

Type: MultiSlot
Units: ENERGY
Description: The sum of the energy consumed by the linked pumped storage reservoirs.
Information:
I/O: Output
Links: Linked to the Pump Energy slot on pumped storage reservoirs.

☞ Pumped Storage Pumping Capacity

Type: MultiSlot
Units: POWER
Description: The sum of the power consumed by the linked pumped storage reservoirs.
Information:
I/O: Output
Links: Linked to the Pump Power slot on pumped storage reservoirs.

☞ Ramping Cost

Type: MultiSlot
Units: $
Description: The sum of the ramping cost at the linked reservoirs.
Information:
I/O: Output
Links: Linked to the Ramping Cost slot on power reservoirs.

☞ Turbine Decrease

Type: MultiSlot
Units: FLOW
Description: The sum of the turbine decrease at the linked reservoirs.
Information:
I/O: Output
Links: Linked to the Turbine Decrease slot on power reservoirs.

☞ Turbine Increase

Type: MultiSlot
Units: FLOW
Description: The sum of the turbine increase at the linked reservoirs.
Information:
I/O: Output
Links: Linked to the Turbine Increase slot on power reservoirs.

☞ System Startup Cost

Type: MultiSlot
Units: VALUE ($)
Description: Total startup cost of all linked reservoirs.
Information: This slot is used to link the Thermal object to each reservoir’s Plant Startup Cost slot.
I/O: Output Only
Links: Yes, to reservoir’s Plant Startup Cost

26.1 User Methods

26.1.1 Load

These methods determine if hourly power load data should be part of the economic valuation of hydropower.

26.1.1.1 None

This default method does not use hourly power load data.

26.1.1.2 Hourly Load

This method calculates the load for each time step as the sum of hourly system load and load adjustment values that are input prior to the beginning of the run.

$$\text{Load}_t = \sum_{\text{hour} \in t} \text{System Load}_{\text{hour}} + \text{Load Adjustment}_{\text{hour}} \quad \forall t$$

The “Thermal” methods use this data for calculating the economic value of hydropower. The “Block Hydro” methods do not use this data.

Slots Specific to this Method:

☞ **Load Adjustment**

Type: Series Slot
Units: POWER
Description: Correction to the specified system load, frequently NaN.
Information:
I/O: Input only
Links: Not linkable.

☞ **System Load**

Type: Series Slot
Units: POWER
Description: The combined load to be met by both thermal power sources and hydropower.
Information:
I/O: Input only
Links: Not linkable.

26.1.2 Modified Load

This category allows the user to define load leveling on the thermal object. The category is dependent on selecting the **Hourly Load** method from the **Load** category.

26.1.2.1 None

This is the default no-action method.

26.1.2.2 Calculate Modified Load

This method takes the System Load (original, unmodified load) and then subtracts the net generation from hydropower to get the Modified Load. It also adds slots to define load leveling in the thermal object. With the new slots, users can write policies that will level the load to the extent possible.

SLOTS SPECIFIC TO THIS METHOD:

☞ **MODIFIED LOAD**

Type: Series Slot
Units: ENERGY
Description: The remaining load after hydropower is generated.
Information: The remaing load equals the original load minus the net generation from the hydropower sources selected in the **Modified Load Power Sources Used** table.
I/O: Output
Links: Not Linkable

☞ **MODIFIED LOAD POWER SOURCES USED**

Type: Table Slot
Units: NONE
Description: One row for each power source and one column with a positive value for power sources to be used.
Information: NaN indicates the power source doesn’t apply or has already been included in reducing the load.
I/O: Input only
Links: Not Linkable

METHOD DETAILS

Like all thermal method calculations, the method is called at the end of a simulation run. This method first calculates the original, unmodified load (as an energy) by adding the hourly system load and load adjustment slots for all hours within a time step.

The user selects the power sources that will be used to mo dify the load by setting values in the **Modified Load Power Sources Used** table. This table has one column and three rows: Conventional Hydro Power Evaluated, Pump Power Evaluated, and Allocated Energy

Evaluated. Power sources that should be used to modify the load have a non-NaN value while unused sources have a NaN value.

The **Modified Load** initially equals the original load (System Load minus Load Adjustment). The net generation from the selected power sources are subtracted from this load. If any of the selected power sources have a value of NaN then the modified load is also NaN. For example, during a normal pre-optimization simulation, modified load will be NaN for all timesteps because hydropower generation is not yet known. In contrast, during a fully specified simulation or a post-optimization rulebased simulation, modified load will be calculated because all hydropower generations will be known.

Allocated power is constrained by minimal and maximal power generation for each time period combined with periodic total energy requirements. The power is allocated to level the load in coordination with other power sources.

After calculating the individual economic components for each time period, the totals for all time periods are placed in the Linear Total Values table slot.

The method is also used in optimization as described HERE (Optimization.pdf, Section 5.10.1.4.2).

26.1.3 Preferred Customer

A group of reservoirs may be obligated to meet the power demands of “preferred customers” before this energy is used for other purposes. For example, if a group of reservoirs is owned by another entity, the energy demand of their customers must be met before the energy is coordinated with the other reservoirs in an economic objective, and in this sense, their customers are “preferred”. In such a situation, some (and perhaps all) of these reservoirs are also treated as “allocated energy” that can be flexibly used in coordination with the remaining reservoirs.

The methods in this category model the energy needs of the preferred customers. The preferred customers are identified by membership in a “Preferred Customer” subbasin. This is a predefined type of subbasin.

Note: Although still shown in the interface, this category no longer has methods that do anything.

26.1.3.1 None

Preferred customers are not used in the economic valuation.

26.1.3.2 Preferred Customer

In simulation, the Preferred Customer Energy slot holds the amount of energy that should be sent to the preferred customers. This energy is subtracted from the allocated energy when evaluating the economic value of hydropower.

In optimization the Preferred Customer Energy is a variable. The optimization adds a constraint that limits the energy consumed by preferred customers to the energy limits of the reservoirs in the Preferred Customer subbasin by automatically adding the following constraint when the preferred customer method is selected:

$$PreferredCustomerEnergy(t) \le$$

$$\sum_{res \in PrefCustSub \cap OptimizedRes} resEnergyMaximum + \sum_{res \in PrefCustSub \cap AllocRes} AllocatedMaximum(res,t) \quad \forall t$$

The first sum is over the subbasin reservoirs that are optimized while the second sum is over the subbasin reservoirs that are not optimized, but instead are part of allocated energy. In the first case, the maximum energy value is accessible through the configuration menu of the energy slot on each reservoir. In the second case, the Allocated Maximum slot on the thermal object holds a value for each time step.

Typically, when preferred customer energy is used, the modeler will add a prioritized constraint to the optimization requiring that the preferred customer energy be met. The Energy demand must be specified on a slot on a data object.

$$PreferredCustomerEnergy(t) \ge DataObjectPreferredCustomerDemand(t) \quad \forall t$$

As in simulation, the Preferred Customer Energy is subtracted from Allocated Energy in the economic evaluation.

Note: Although still shown in the interface, this method no longer does anything.

SLOTS SPECIFIC TO THIS METHOD:

☞ **PREFERRED CUSTOMER ENERGY**

Type: Agg Series Slot
Units: ENERGY
Description: Load for preferred customers stated as an energy.
Information:
I/O: Input or Output
Links: Not usually linked.

26.1.4 Regulation

Regulation is one of the ancillary services that hydropower plants can supply in addition to power to increase the reliability and flexibility of the power system to adjust to fluctuations in power demand and supply. When a plant is regulating it will follow the load within some prescribed band of power rather than generating a fixed amount of power. Typically, regulation is a valuable service that can be provided efficiently by hydropower compared to alternative power sources. In locations where regulation is marketed it usually commands a solid premium above the price of the power generated. This value is partially reduced by the increased maintenance costs associated with regulation.

There are three methods in this category: “One Sided Regulation”, “Two sided Regulation” and “One and Two Sided Regulation”. The design of one sided regulation is almost identical to two sided regulation except that there are separate calculations for regulation up and regulation down. The “One and Two Sided Regulation” method will allow the user to use both approaches within a single model.

26.1.4.1 None

Regulation services are not evaluated.

26.1.4.2 Two Sided Regulation

Regulation services are valued as a piecewise linear function of the amount of regulation provided minus the additional operating costs (wear and tear, maintenance, etc.) associated with regulation. The piecewise linear valuation is analogous to the block hydro method of valuing generation (described in “Preferred Customer” HERE (Section 26.1.3)) All of the associated slots for this method have a compatible slot in the block hydro method. The operating costs are simply the sum of individual operating costs calculated on the power reservoirs that are providing regulation.

Two sided regulation assumes that frequency regulation is scheduled with an equal amount of regulation up and regulation down available, and a single valuation for this ancillary service. In contrast, one sided regulation allows regulation up and regulation down to be separate ancillary services with separate amounts scheduled and usually valued differently.

The System Regulation slot holds the total two sided regulation for all power reservoirs linked to it. The value of marginal value of regulation is a non-increasing function of the amount of regulation. In RiverWare, this function is modeled as a series of discrete blocks with decreasing (or equal) marginal values of regulation. These values are in the Regulation Block Costs slot. The amount of each block used in the valuation is in the Block Regulation slot. These blocks are used in order of most valuable to least valuable.

The Regulation Marginal Value and Previous Regulation Marginal Value slots hold the marginal value of increasing and decreasing regulation respectively. If the valuation partially uses a block, then both the Regulation Marginal Value and Previous Regulation Marginal Value slots will have the value of that block. If instead, say N blocks are fully used, and Block N+1 is unused, then Regulation Marginal Value will have the value of Block N+1 and Previous Regulation Marginal Value will have the value of Block N.

This method requires either Calculate Linear Economic Value or Calculate Block Economic Value method to be selected.

Simulation and Optimization use the following equation to calculate the value of regulation by summing over blocks:

$$RegulationValue(t) = \sum_{blocks} RegulationBlockCost(block,t) \times RegulationEnergy(block,t) \quad \forall t$$

The energy in the regulation blocks is constrained to equal the total system regulation (converted from power to energy by multiplying by the timestep).

$$SystemRegulation(t) \times SecondsPerTimestep(t) = \sum_{blocks} RegulationEnergy(block,t) \quad \forall t$$

Slots Specific to this Method:

☞ Block Regulation

Type: Agg Series Slot
Units: Energy
Description: Each column is the use of a regulation block in the new optimization system. The slot maximum value is used for the block size. The total equals the System Regulation.
Information: Used in the optimization formulation of regulation.
I/O: Output
Links: Not linked.

☞ Marginal Regulation Costs

Type: Agg Series Slot
Units: $/Power
Description: The value of one additional unit of regulation.
Information:
I/O: Output
Links: Not linked.

☞ Previous Marginal Regulation Costs

Type: Series Slot

Units: $/POWER
Description: The cost of reducing regulation by one unit.
Information:
I/O: Output
Links: Not linked.

☞ REGULATION BLOCK CAPABILITIES
Type: Table Series Slot
Units: POWER
Description: The size of the each regulation block.
Information: The length of the piecewise linear segments in the value of regulation function.
I/O: Input
Links: Not linkable.

☞ REGULATION BLOCK COSTS
Type: Agg Series Slot
Units: $/POWER
Description: The value of regulation for each regulation block.
Information: The slope of the piecewise linear segments in the value of regulation function.
I/O: Input
Links: Not linkable.

☞ REGULATION ENERGY
Type: Agg Series Slot
Units: ENERGY
Description: This slot has one column for each block of regulation. The values represent the energy in each block that was used to value the regulation generated during the run. At most one block will have a value that is strictly between zero and the size of the block, the Regulation Block Capability. The preceding blocks will be used entirely and the regulation energy will equal the block size. The succeeding blocks will be unused.
Information:
I/O: Output
Links: Not linked.

☞ REGULATION VALUE
Type: Agg Series Slot
Units: $
Description: The calculated value of regulation based on the Regulation Block Costs.
Information:
I/O: Output
Links: Not linked.

☞ **SYSTEM OPERATING COST**

Type: MultiSlot
Units: $
Description: The sum of operating costs from regulation for all linked reservoirs.
Information:
I/O: Output
Links: Linked to Operating Cost on power reservoirs.

☞ **SYSTEM REGULATION**

Type: MultiSlot
Units: POWER
Description: The total regulation summed from the linked power reservoirs.
Information:
I/O: Output
Links: Linked to Regulation on power reservoirs.

26.1.4.3 One Sided Regulation method

The value of one sided regulation is philosophically similar to two sided regulation. The only difference is that regulation up and regulation down have separate slots and separate valuations. Physically, this may lead either to solutions that use only one side of regulation or solutions that use both sides, but perhaps unevenly. For one sided regulation, there are two slots for every analogous slot in two sided regulation, an “Up” slot and a “Down” slot. Similarly, the System Regulation Up and System Regulation Down slots should be linked to Regulation Up and Regulation Down respectively on power reservoirs. For more detail on the specific calculations, see the description of two sided regulation.

Note: Although the method can be selected and slots will be instantiated, this method has **not** been implemented yet.

SLOTS ADDED BY THIS METHOD

The following slots are instantiated when this method is selected:

☞ **SYSTEM REGULATION UP**

Type: MultiSlot
Units: POWER
Description: Total system regulation up
Information: This slot is used to link the Thermal object to each reservoir’s Plant Regulation Up slot.
I/O: Output only
Links: Linkable to Plant Regulation Up

☞ Block Regulation Up

Type: AggSeriesSlot
Units: ENERGY
Description: The blocks of energy used in the valuation.
Information: The sum of these blocks equals the System Regulation Up. The value of each block is given in Regulation Up Block Costs.
I/O: Output
Links: not linkable

☞ Regulation Up Block Costs

Type: AggSeriesSlot
Units: POWERCOST
Description: The data to value each block of regulation up.
Information:
I/O: Input
Links: Not linkable

☞ Regulation Up Value

Type: SeriesSlot
Units: VALUE ($)
Description: The total value of regulating up.
Information: It is calculated as the sum of the value of the blocks used
I/O: output
Links:

☞ Regulation Up Marginal Value

Type: SeriesSlot
Units: POWERCOST
Description: This is the incremental value of the regulating up block being used.
Information: If regulation is between blocks, this is the value of the next block. This value is only calculated in simulation.
I/O: Output only
Links: NA

☞ Regulation Up Previous Marginal Value

Type: SeriesSlot
Units: POWERCOST
Description: This is the incremental value of the regulating up block being used.
Information: If regulation is between blocks, this is the value of the previous block. This value is only calculated in simulation.
I/O: Output only
Links: NA

☞ System Regulation Down

Type: MultiSlot
Units: POWER
Description: Total system regulation down
Information: This is used to link the Thermal object to each reservoir's Plant Regulation Down slot.
I/O: Output only
Links: Linkable to each reservoir's Plant Regulation Down

☞ Block Regulation Down

Type: AggSeriesSlot
Units: ENERGY
Description: The blocks of energy used in the valuation.
Information: The sum of these blocks equals the System Regulation Down. The value of each block is given in Regulation Down Block Costs.
I/O: Output
Links: NA

☞ Regulation Down Block Costs

Type: AggSeriesSlot
Units: POWERCOST
Description: This is the data to value each block of regulation down.
Information:
I/O:
Links: Not linked

☞ Regulation Down Value

Type: SeriesSlot
Units: VALUE ($)
Description: the total value of regulating down
Information: This is equal to the sum of the value of the blocks used.
I/O: Output
Links: Not linkable

☞ Regulation Down Marginal Value

Type: SeriesSlot
Units: POWERCOST
Description: The incremental value of the regulating down block being used.
Information: If regulation is between blocks, this is the value of the next block. This value is only calculated in simulation.
I/O: Output only
Links: Not linkable

☞ **Regulation Down Previous Marginal Value**

Type: SeriesSlot
Units: PowerCost
Description: The incremental value of the regulating down block being used.
Information: If regulation is between blocks, this is the value of the previous block. This value is only calculated in simulation.
I/O: output only
Links:

☞ **System Operating Cost**

Type: MultiSlot
Units: Value ($)
Description: Total system operating cost
Information: This used to link the Thermal object to each reservoir's Operating Cost slot.
I/O: Output
Links: Linkable to reservoir Operating Cost slot

Method Details

The design of One Sided Regulation is almost identical to Two Sided Regulation except that there are separate calculations for regulation up and regulation down.

- Regulation Up Value = Sum over blocks (Regulation Up Block Costs * Block Regulation Up)
- Regulation Down Value = Sum over blocks (Regulation Down Block Costs * Block Regulation Down)

26.1.4.4 One and Two Sided Regulation method

This method allows the user to perform both one and two sided regulation in a single model. The slots added by this method include all of those for one sided regulation listed above and those for two sided regulation listed in the online help. This method will simply call the two sided regulation method, then call the one sided regulation method.

Note: Although the method can be selected and slots will be instantiated, this method has **not** been implemented yet.

26.1.5 Linear Economic Value

26.1.5.1 None

No evaluation is made.

26.1.5.2 Calculate Linear Economic Value

This method applies a time dependent linear value to power generated. This method is also known as POSE or System Lambda. The order of power source evaluation doesn't matter, only which power sources are being evaluated, because of the linear valuation. In general, the calculation is a simplified version of the existing block calculation. The value of hydropower is in the Linear Hydro Costs series slot.

The power sources to use in the evaluation are selected in the Linear Power Sources Used table slot. The possible power sources are conventional hydropower, pumped storage, and allocated power. The calculated value of each type of power is output to a corresponding series slot.

Allocated power is constrained by minimal and maximal power generation for each time period combined with periodic total energy requirements. The power is allocated for maximal economic gain in coordination with other power sources.

After calculating the individual economic components for each time period, the totals for all time periods are placed in the Linear Total Values table slot.

Slots Associated with the Calculate Linear Economic Value Method

☞ **Linear Hydro Costs**

Type: Series Slot
Units: PowerCost
Description: The value of each unit of hydropower generated.
Information: This method does not have Marginal Operating Cost or Previous Marginal Operating Cost slots because the values would be the same as this slot.
I/O: Input
Links: Not Linkable

☞ **Linear Power Sources Used**

Type: Table Slot
Units: None
Description: One row for each power source and one column with a positive value for power sources to be used.
Information: NaN indicates the power source doesn't apply or has already been included in economic calculations.
I/O: Input
Links: Not Linkable

☞ LINEAR CONVENTIONAL REPLACEMENT VALUE

Type: Series Slot
Units: NONE
Description: The calculated value of conventional hydropower using linear evaluation.
Information:
I/O: Output
Links: Not Linkable

☞ LINEAR PUMP REPLACEMENT VALUE

Type: Series Slot
Units: VALUE
Description: The calculated value of net hydropower generated by pumps using linear evaluation.
Information:
I/O: Output
Links: Not Linkable

☞ LINEAR ALLOCATED REPLACEMENT VALUE

Type: Series Slot
Units: VALUE
Description: The calculated value of allocated hydropower using linear evaluation.
Information:
I/O: Output
Links: Not Linkable

☞ LINEAR AVOIDED OPERATING COST

Type: Series Slot
Units: VALUE
Description: The combined value of conventional, pump, and allocated hydropower using linear evaluation.
Information:
I/O: Output
Links: Not Linkable

☞ LINEAR NET AVOIDED COST

Type: Series Slot
Units: VALUE
Description: Linear Avoided Operating Cost minus the value of water used for spill or generation.
Information:
I/O: Output
Links: Not Linkable

☞ Linear Total Values

Type: Table Slot
Units: NONE
Description: The sum over time for several slots used in the linear economic evaluation.
Information: One column. One row for each slot:

- Conventional Hydro Power
- Pump Power Evaluated
- Allocated Energy Evaluated
- Conventional Total Avoided Cost
- Pump Total Avoided Cost
- Allocated Total Avoided Cost
- Total Avoided Operating Cost All
- Regulation Value
- Reg. Operating Cost
- Total Future Value of Used Energy
- Total Spill Cost
- Cumul Stor Value
- Net Total Avoided Cost
- Objective

I/O: Output
Links: Not Linkable

☞ Allocated Minimum

Type: Agg Series Slot
Units: POWER
Description: The minimum power generation for projects with allocated energy.
Information: One column for each project to be allocated.
I/O: Input
Links: Not Linkable

☞ Allocated Maximum

Type: Agg Series Slot
Units: POWER
Description: The maximum power generation for projects with allocated energy.
Information: One column for each project to be allocated.
I/O: Input
Links: Not Linkable

☞ Allocated Total Energy

Type: Agg Series Slot

Units: ENERGY
Description: The total energy to be allocated from a project over some time period.
Information: One column for each project to be allocated. Totals should be entered with some consistent period. For example, values entered only once a day will be allocated across that day assuming the model has a timestep smaller than one day.
I/O: Input
Links: Not Linkable

☞ **ALLOCATED DETAIL ENERGY**
Type: Agg Series Slot
Units: ENERGY
Description: The result of allocating the energy to individual time periods.
Information: One column for each project to be allocated.
I/O: Output
Links: Not Linkable

☞ **ALLOCATED ENERGY SUM**
Type: Series Slot
Units: ENERGY
Description: The sum of allocated energy across projects.
Information: Only used in optimization to assist in mathematical representation.
I/O: Output
Links: Not Linkable

26.1.6 Block Economic Value

26.1.6.1 None

No evaluation is made.

26.1.6.2 Calculate Block Economic Value

This method applies a time dependent piecewise linear, or block, value to power generated. The order of power source evaluation matters for simulation because it is nonlinear; the order affects the “credit” given to each power source. Evaluation before other power sources will result in higher credit. The optimization doesn’t allocate credit to individual power sources, and the order has no effect on it.

At the end of the run, the economic value of hydropower is calculated based on a piecewise linear function of power. The piecewise function can vary by time period. While there is no direct linkage to thermal sources of power, one interpretation of the piecewise linear function is that it represents the coordinated reduction in other power sources as a result of hydropower generation. This method does not attempt to meet an explicit load, rather the load is assumed to be included in the process that generates the piecewise linear functions.

The piecewise linear functions are represented by the Hydro Block Costs. The Hydro Block Costs are the slope of the piecewise linear segments, or alternatively the value of a unit of power. The piecewise linear function is assumed to be concave: the costs are assumed to decrease as the block number increases.

Slots Associated with the Calculate Block Economic Value Method

☞ Hydro Block Costs

Type: Agg Series Slot
Units: PowerCost
Description: The value of each unit of hydropower generated.
Information: The columns represent the power blocks which are assumed to have a non increasing value for power. The number of columns with valid values should equal the Number of Hydro Blocks scalar slot value.
I/O: Input
Links: Not Linkable

☞ Hydro Block Use

Type: Agg Series Slot
Units: Energy
Description: The blocks used to value hydropower.
Information: The sum of the blocks should equal the hydropower generated unless too few blocks are specified. The user must input the size of the blocks. The slot configuration dialog should be used to set the Minimum to zero and the Maximum to the desired block size. The Maximum values replace the previous Hydro Block Capacity slot. Maximum values must be entered for at least the same number of columns specified in the Number of Hydro Blocks scalar slot.
I/O: Input
Links: Not Linkable

☞ Number of Hydro Blocks

Type: Scalar Slot
Units: None
Description: The maximum number of hydro blocks that could potentially be used.
Information: This value should equal the number of columns in Hydro Block Costs with valid values. If the number in this slot is greater than the number of columns in Hydro Block Costs with valid values, the run will abort with an error message.
I/O: If not input it will default to the total number of columns in Hydro Block Costs.

Links: Not Linkable

☞ Block Power Source Evaluation Order

Type: Table Slot
Units: None
Description: One row for each power source and one column with a unique positive value for power sources to be used.
Information: NaN indicates the power source doesn't apply or has already been included in economic calculations.
I/O: Input
Links: Not Linkable

☞ Block Marginal Operating Cost

Type: Series Slot
Units: PowerCost
Description: The additional marginal value of an extra unit of hydropower.
Information:
I/O: Output
Links: Not Linkable

☞ Block Previous Marginal Operating Cost

Type: Series Slot
Units: PowerCost
Description: The additional marginal value of the previous unit of unit of hydropower generated.
Information: This value will be the same as the Block Marginal Operating Cost if hydrogeneration is in the middle of a block.However, the hydro generation for some time periods may exactly match the full use of some number of blocks. This is particularly likely for optimization models. In these cases the marginal operating cost and the previous marginal operating cost will differ because they represent different blocks.
I/O: Output
Links: Not Linkable

☞ Block Tolerance

Type: Scalar Slot
Units: Power
Description: The tolerance for considering a solution to have hydrogeneration exactly equal to some number of blocks.
Information: Floating point arithmetic usually requires a tolerance for comparing numbers. In this case, if the values are equal within the tolerance, then the Block Previous Marginal Operating Cost will have a different value than the Block Marginal Operating Cost. This slot used to be a Table Slot.

I/O: Input
Links: Not Linkable

☞ Block Conventional Replacement Value

Type: Series Slot
Units: NONE
Description: The calculated value of conventional hydropower using block evaluation.
Information:
I/O: Output
Links: Not Linkable

☞ Block Pump Replacement Value

Type: Series Slot
Units: VALUE
Description: The calculated value of net hydropower generated by pumps using block evaluation.
Information:
I/O: Output
Links: Not Linkable

☞ Block Allocated Replacement Value

Type: Series Slot
Units: VALUE
Description: The calculated value of allocated hydropower using block evaluation.
Information:
I/O: Output
Links: Not Linkable

☞ Block Avoided Operating Cost

Type: Series Slot
Units: VALUE
Description: The combined value of conventional, pump, and allocated hydropower using block evaluation.
Information:
I/O: Output
Links: Not Linkable

☞ Block Net Avoided Cost

Type: Series Slot
Units: VALUE
Description: Block Avoided Operating Cost minus the value of water used for spill or generation.
Information:

I/O: Output
Links: Not Linkable

☞ BLOCK TOTAL VALUES

Type: Table Slot
Units: NONE
Description: The sum over time for several slots used in the block economic evaluation.
Information: One column. One row for each slot:

- Conventional Hydro Power
- Pump Power Evaluated
- Allocated Energy Evaluated
- Conventional Total Avoided Cost
- Pump Total Avoided Cost
- Allocated Total Avoided Cost
- Total Avoided Operating Cost All
- Regulation Value
- Reg. Operating Cost
- Total Future Value of Used Energy
- Total Spill Cost
- Cumul Stor Value
- Net Total Avoided Cost
- Objective

I/O: Output
Links: Not Linkable

☞ ALLOCATED MINIMUM

Type: Agg Series Slot
Units: POWER
Description: The minimum power generation for projects with allocated energy.
Information: One column for each project to be allocated.
I/O: Input
Links: Not Linkable

☞ ALLOCATED MAXIMUM

Type: Agg Series Slot
Units: POWER
Description: The maximum power generation for projects with allocated energy.
Information: One column for each project to be allocated.
I/O: Input
Links: Not Linkable

☞ **ALLOCATED TOTAL ENERGY**

Type: Agg Series Slot
Units: ENERGY
Description: The total energy to be allocated from a project over some time period.
Information: One column for each project to be allocated. Totals should be entered with some consistent period. For example, values entered only once a day will be allocated across that day assuming the model has a timestep smaller than one day.
I/O: Input
Links: Not Linkable

☞ **ALLOCATED DETAIL ENERGY**

Type: Agg Series Slot
Units: ENERGY
Description: The result of allocating the energy to individual time periods.
Information: One column for each project to be allocated.
I/O: Output
Links: Not Linkable

☞ **ALLOCATED ENERGY SUM**

Type: Series Slot
Units: ENERGY
Description: The sum of allocated energy across projects.
Information: Only used in optimization to assist in mathematical representation.
I/O: Output
Links: Not Linkable

26.1.7 Thermal Unit Replacement Value

26.1.7.1 None

No evaluation is made.

26.1.7.2 Calculate Thermal Unit Replacement Value

This method applies a time dependent piecewise linear value to power generated. The value is based on replacing thermal units. The order of power source evaluation matters for simulation because it is nonlinear; the order affects the “credit” given to each power source. Evaluation before other power sources will result in higher credit. The optimization doesn’t allocate credit to individual power sources, and the order has no effect on it.

At the end of the run, the economic value of hydropower is calculated based on the savings from replacing generation from thermal power sources. The first part of the calculation is to assign thermal power sources to meet a load based on their average cost of operation. The second part of the calculation is to use hydropower to back down generation from the least efficient units that were used to meet the load in the initial calculation.

This method is dependent on having Hourly Load method selected.

Slots Associated with the Calculate Thermal Unit Replacement Value Method

☞ Thermal Unit Cost

Type: Agg Series Slot
Units: PowerCost
Description: The cost of generating at each unit.
Information: The columns represent the thermal units.
I/O: Input
Links: Not Linkable

☞ Thermal Unit Capacity

Type: Table Slot
Units: Power
Description: The size of the units used to value hydropower.
Information:
I/O: Input
Links: Not Linkable

☞ Thermal Unit Availability

Type: Agg Series Slot
Units: No units
Description: The availability of each unit for each time period.
Information: A fraction ranging from 0 to 1.
I/O: Input
Links: Not Linkable

☞ Thermal Energy

Type: Agg Series Slot
Units: Energy
Description: The energy generated by each unit in order to meet the load.
Information: One column for each unit.
I/O: Output
Links: Not Linkable

☞ Thermal Power Source Evaluation Order

Type: Table Slot
Units: None
Description: One row for each power source and one column with a unique positive value for power sources to be used.

Information: NaN indicates the power source doesn't apply or has already been included in economic calculations.
I/O: Input
Links: Not Linkable

☞ Thermal Marginal Operating Cost

Type: Series Slot
Units: PowerCost
Description: The additional marginal value of an extra unit of hydropower.
Information:
I/O: Output
Links: Not Linkable

☞ Thermal Previous Marginal Operating Cost

Type: Series Slot
Units: PowerCost
Description: The additional marginal value of the previous unit of unit of hydropower generated.
Information: This value will be the same as the Thermal Marginal Operating Cost if hydrogeneration is in the middle of a unit. However, the hydro generation for some time periods may exactly match the full use of some number of units. This is particularly likely for optimization models. In these cases the marginal operating cost and the previous marginal operating cost will differ because they represent different units.
I/O: Output
Links: Not Linkable

☞ Thermal Tolerance

Type: Scalar Slot
Units: Power
Description: The tolerance for considering a solution to have hydrogeneration exactly equal to some number of units.
Information: Floating point arithmetic usually requires a tolerance for comparing numbers. In this case, if the values are equal within the tolerance, then the Thermal Previous Marginal Operating Cost will have a different value than the Thermal Marginal Operating Cost.
I/O: Input
Links: Not Linkable

☞ Thermal Only Cost

Type: Series Slot
Units: PowerCost
Description: The cost of meeting the loads with only thermal units and no hydropower.

Information: One column. One row for each slot:

- Conventional Hydro Power
- Pump Power Evaluated
- Allocated Energy Evaluated
- Conventional Total Avoided Cost
- Pump Total Avoided Cost
- Allocated Total Avoided Cost
- Total Avoided Operating Cost All
- Regulation Value
- Reg. Operating Cost
- Total Future Value of Used Energy
- Total Spill Cost
- Cumul Stor Value
- Net Total Avoided Cost
- Objective

I/O: Output
Links: Not Linkable

☞ Thermal Conventional Replacement Value

Type: Series Slot
Units: NONE
Description: The calculated value of conventional hydropower using thermal evaluation.
Information:
I/O: Output
Links: Not Linkable

☞ Thermal Pump Replacement Value

Type: Series Slot
Units: VALUE
Description: The calculated value of net hydropower generated by pumps using thermal evaluation.
Information:
I/O: Output
Links: Not Linkable

☞ Thermal Allocated Replacement Value

Type: Series Slot
Units: VALUE
Description: The calculated value of allocated hydropower using thermal evaluation.
Information:

I/O: Output
Links: Not Linkable

☞ Thermal Avoided Operating Cost

Type: Series Slot
Units: Value
Description: The combined value of conventional, pump, and allocated hydropower using thermal evaluation.
Information:
I/O: Output
Links: Not Linkable

☞ Thermal Net Avoided Cost

Type: Series Slot
Units: Value
Description: Thermal Avoided Operating Cost minus the value of water used for spill or generation.
Information:
I/O: Output
Links: Not Linkable

☞ Thermal Total Values

Type: Table Slot
Units: None
Description: The sum over time for several slots used in the thermal economic evaluation.
Information: One row for each slot. One column.
I/O: Output
Links: Not Linkable

☞ Allocated Minimum

Type: Agg Series Slot
Units: Power
Description: The minimum power generation for projects with allocated energy.
Information: One column for each project to be allocated.
I/O: Input
Links: Not Linkable

☞ Allocated Maximum

Type: Agg Series Slot
Units: Power
Description: The maximum power generation for projects with allocated energy.
Information: One column for each project to be allocated.
I/O: Input

Links: Not Linkable

☞ **ALLOCATED TOTAL ENERGY**

Type: Agg Series Slot
Units: ENERGY
Description: The total energy to be allocated from a project over some time period.
Information: One column for each project to be allocated. Totals should be entered with some consistent period. For example, values entered only once a day will be allocated across that day assuming the model has a timestep smaller than one day.
I/O: Input
Links: Not Linkable

☞ **ALLOCATED DETAIL ENERGY**

Type: Agg Series Slot
Units: ENERGY
Description: The result of allocating the energy to individual time periods.
Information: One column for each project to be allocated.
I/O: Output
Links: Not Linkable

☞ **ALLOCATED ENERGY SUM**

Type: Series Slot
Units: ENERGY
Description: The sum of allocated energy across projects.
Information: Only used in optimization to assist in mathematical representation.
I/O: Output
Links: Not Linkable

26.1.8 Linear Objective Reported

These methods are used to report the simulation calculation of an objective function value used in optimization. The Linear Objective Reported category is dependent on selecting the Calculate Linear Economic Value evaluation method. The “objective” value reported consists of a diagnostic message and is also stored in the Linear Total Values table slot. The methods have no dependent slots.

26.1.8.1 None

No diagnostic message is printed for the objective.

26.1.8.2 Avoided Cost

The Linear Avoided Operating Cost is reported.

26.1.8.3 Avoided Cost Plus Cumulative Storage Value

The combined value of the ending Total Cumulative Storage plus the Linear Avoided Operating Cost is reported.

26.1.8.4 Avoided Cost Plus Net Cumulative Storage Value

The combined value of the net increase in Total Cumulative Storage during the run plus the Linear Avoided Operating Cost is reported. The net increase equals the Total Cumulative Value of Storage on the last time period of the run minus the Total Cumulative Value of Storage at the starting time step.

26.1.8.5 Avoided Cost Plus Net CSV Plus Net Regulation

The combined value of the net increase in Total Cumulative Storage during the run, the Linear Avoided Operating Cost, and the net value of frequency regulation (minus regulation operating costs) is reported.

26.1.9 Block Objective Reported

These methods are used to report the simulation calculation of an objective function value used in optimization. The Block Objective Reported category is dependent on selecting the Calculate Block Economic Value evaluation method. The “objective” value reported consists of a diagnostic message and is also stored in the Block Total Values table slot. The methods have no dependent slots.

26.1.9.1 None

No diagnostic message is printed for the objective.

26.1.9.2 Avoided Cost

The Block Avoided Operating Cost is reported.

26.1.9.3 Avoided Cost Plus Cumulative Storage Value

The combined value of the ending Total Cumulative Storage plus the Block Avoided Operating Cost is reported.

26.1.9.4 Avoided Cost Plus Net Cumulative Storage Value

The combined value of the net increase in Total Cumulative Storage during the run plus the Block Avoided Operating Cost is reported. The net increase equals the Total Cumulative Value of Storage on the last time period of the run minus the Total Cumulative Value of Storage at the starting time step.

26.1.9.5 Avoided Cost Plus Net CSV Plus Net Regulation

The combined value of the net increase in Total Cumulative Storage during the run, the Block Avoided Operating Cost, and the net value of frequency regulation (minus regulation operating costs) is reported.

26.1.10 Thermal Objective Reported

These methods are used to report the simulation calculation of an objective function value used in optimization. The Thermal Objective Reported category is dependent on selecting the Calculate Thermal Economic Value evaluation method. The "objective" value reported consists of a diagnostic message and is also stored in the Thermal Total Values table slot. The methods have no dependent slots.

26.1.10.1 None

No diagnostic message is printed for the objective.

26.1.10.2 Avoided Cost

The Thermal Avoided Operating Cost is reported.

26.1.10.3 Avoided Cost Plus Cumulative Storage Value

The combined value of the ending Total Cumulative Storage plus the Thermal Avoided Operating Cost is reported.

26.1.10.4 Avoided Cost Plus Net Cumulative Storage Value

The combined value of the net increase in Total Cumulative Storage during the run plus the Thermal Avoided Operating Cost is reported. The net increase equals the Total Cumulative Value of Storage on the last time period of the run minus the Total Cumulative Value of Storage at the starting time step.

26.1.10.5 Avoided Cost Plus Net CSV Plus Net Regulation

The combined value of the net increase in Total Cumulative Storage during the run, the Thermal Avoided Operating Cost, and the net value of frequency regulation (minus regulation operating costs) is reported.

27. Water User

The Water User object can exist as a separate "Stand Alone" object on the workspace as well as a member element of the Aggregate Diversion Site (Agg Diversion Site) object. The Water User simulates the consumption/depletion of water often from a reservoir or reach.

When the Water User object is an element of an Agg Diversion Site, the Link Structure selected on the Agg Diversion Site determines how the Water User elements interact with each other, the Agg Diversion Site, and the object from which water is being diverted (i.e. reach or reservoir). When the Water User is used as a single object on the workspace, it functions identically to an element on an Agg Diversion Site using the default (No Structure) linking structure. The three linking structures exist on the Agg Diversion Site object.

The general slots which exist on the Water User regardless of the type of linking structure that exists on the Agg Diversion Site are:

General Slots

SLOTS WHICH ALWAYS APPEAR FOR THIS OBJECT

☞ DEPLETION REQUESTED

Type: Series Slot
Units: FLOW
Description: amount of water to be consumed, given the request is met
Information: Automatically linked to the Total Depletion Requested multislot on the Agg Diversion Site.
I/O: Input, Output, or set by a rule
Links: Usually not linked

☞ DIVERSION REQUESTED

Type: Series Slot
Units: FLOW
Description: amount of water requested by the Water User
Information: Automatically linked to the Total Diversion Requested multislot on the Agg Diversion Site.
I/O: Input, Output, or set by a rule
Links: Usually not linked

☞ WATER USER GROUPS

Type: List Slot

Units: NA
Description: Contains the user-defined subbasins to which this Water User is a member. It can be used to encode and display attributes of the Water User. For example, a water user may only be able to use "Imported Water" while other water users may only use "Basin Water". By defining a subbasin for "Imported Water" and one for "Basin Water" and adding the appropriate water users to each, you can keep summarize information about each group of users. Perhaps you wish to create an expression slot that sums up the total diversion requested for Imported Water. You can use the "Imported Water" subbasin. The Water User Groups slot then allows you to see, from each water user, the groups to which it belongs.
Information: This slot is a special type of list slot called "Subbasin Membership List Slot". The values in the slot are the subbasins to which this water user is a member. To add/remove an entry from the slot, you must go to the subbasin manager and add/remove the water user from the particular subbasin. The slot does provide a menu option to open the subbasin manager File->Edit Subbasins.
I/O: Specified by subbasin membership.
Links: Not Linkable

Stand Alone or No Structure Slots

In addition to the general slots above, the following slots are shown for a stand alone water user or when the parent Agg Diversion Site has the **No Structure** selected:

(Note that the **No Structure** method is the default linking structure)

☞ DEPLETION
Type: Series Slot
Units: FLOW
Description: amount of water consumed by the water user
Information:
I/O: Output only
Links: Usually not linked

☞ DEPLETION SHORTAGE
Type: Series Slot
Units: FLOW
Description: the difference between the Depletion Requested and the Depletion
Information: This value is only calculated if Depletion Requested and Depletion are both valid.
I/O: Output only
Links: Usually not linked

☞ **DIVERSION**

Type: Series Slot
Units: FLOW
Description: amount of water actually taken (diverted) by the water user
Information:
I/O: Optional; Can be input or calculated based on Diversion Requested and Incoming Available Water (see dispatch methods).
Links: Must be linked to the Diversion slot on the object being diverted from if using the No Structure.

☞ **DIVERSION SHORTAGE**

Type: Series Slot
Units: FLOW
Description: the difference between the Diversion Requested and the Diversion
Information:
I/O: Output only
Links: Usually not linked

☞ **INCOMING AVAILABLE WATER**

Type: Series Slot
Units: FLOW
Description: water available to be diverted
Information: It is the user's responsibility to link this slot to the appropriate slot on the object from which water is diverted if using No Structure.
I/O: Set via link
Links: Must be linked to the Available for Diversion slot on the object being diverted from if using No Structure. If using the sequential structure it is automatically linked to the Agg Diversion Site Total Diversion.

Lumped Structure Slots

In addition to the general slots above, the following slots are shown when the parent Agg Diversion Site has the **Lumped Structure** selected:

☞ **DEPLETION SHORTAGE**

Type: Series Slot
Units: FLOW
Description: The amount of the Depletion Request that could not be met.
Information: This value is calculated by the Agg Diversion Site HERE (Section 3.2.4) when it dispatched. Depletion Requested must be valid.
I/O: Output only
Description: Usually not linked

Sequential Structure Slots

In addition to the general slots above, the following slots are shown when the parent Agg Diversion Site has the **Sequential** structure selected:

☞ **OUTGOING AVAILABLE WATER**

Type: Series Slot
Units: FLOW
Description: portion of the Incoming Available Water that is unused by the Water User
Information:
I/O: Output only
Links: Automatically linked to the Incoming Available Water slot on the next Water User or to the Total Unused Water slot on the Agg Diversion Site if it is the last Water User.

When the sequential linking structure is active, the Incoming Available Water slot on the first Water User element is automatically linked to the Total Diversion slot on the Agg Diversion Site.

27.1 User Methods

The user methods categories available on the Water User object are dependent upon the linking structure selected on the Agg Diversion Site object. The Water User object only dispatches when either the No Structure or Sequential Structure linking structures are selected on the AggDiversion Site. Therefore, return flow categories are only available when one of these two linking structures are selected. If Lumped Structure is selected, the Water User object simply contains the values of Depletion Requested and Diversion Requested that are used by the Agg Diversion Site. These values can be calculated by the Water User based on agricultural data.

When the Water User is not used as an element in the Agg Diversion Site, it has the same user methods as an element in the No Structure linking structure.

27.1.1 Diversion and Depletion Request

This user method category is common to all three linking structures. The methods available in this category are used to calculate the Water User's **Diversion Requested** and **Depletion Requested**. These slots are either set by user input or calculated based on the methods described below.

Note, these methods are executed at different times depending on the method:

- **Reservoir Level Lookup** is executed at the beginning of the timestep.
- **Irrigation Requests with Soil Moisture** is executed partially at the beginning of the timestep and partially in the dispatch method.
- All other methods are executed at beginning of run. As a result, any input data will be used in the calculations as described. If you plan to use a rule to set either **Diversion Requested** or **Depletion Requested**, the other one will not be calculated using these methods as it will not be known at the beginning of the run. In this situation, the rule must set **Diversion Requested** to allow the water user to dispatch and solve. **Depletion Requested**, if not specified, will be set equal to Diversion Requested for internal calculations but will not be set on the slot.

27.1.1.1 Input Requests

This method is common for all linking structures. It is used when the user wants to manually set the values of Diversion Requested and/or Depletion Requested. This is the default method for the Diversion and Depletion Request category.

Slots Specific to this Method

☞ Minimum Diversion Request

Type: Table Slot
Units: FLOW
Description: minimum possible flow rate needed to meet depletion requests
Information: The Minimum Diversion Request can be based on either physical, legal, or environmental restrictions, etc. This is only used if the Depletion Request is non-zero.
I/O: Optional.
Links: Not linkable

No calculations are performed with this method. The user can specify either Diversion Request or Depletion Requested or both. If only Diversion Requested is known, Depletion Requested is set equal to Diversion Requested. If only Depletion Requested is known, Diversion Requested is set equal to Depletion Requested. If Depletion Requested and Minimum Diversion Request are known and Diversion Requested is not, Diversion Requested is set equal to the maximum of Depletion Requested and Minimum Diversion Requested.

See the note **HERE** for information on how this method behaves in Rulebased Simulation.

27.1.1.2 Input Diversion Requests

The Input Diversion Requests method computes Depletion Requested based on an input Diversion Requested.

Slots Specific to this Method

☞ Minimum Efficiency

Type: Series Slot
Units: FRACTION
Description: minimum amount of consumed flow per unit diverted flow
Information: Expressed as a fraction
I/O: Required input
Links: Not linkable

When Diversion Requested is input, Depletion Requested is computed with the following equation:

$$DepletionRequested = DiversionRequested \bullet MinimumEfficiency$$

See the note **HERE** for information on how this method behaves in Rulebased Simulation.

27.1.1.3 Input Depletion Requests

The Input Depletion Requests method computes Diversion Requested based on an input Depletion Requested.

Slots Specific to this Method

☞ Minimum Efficiency

Type: Series Slot
Units: Fraction
Description: minimum amount of consumed flow per unit diverted flow
Information: Expressed as a fraction
I/O: Required input
Links: Not linkable

☞ Maximum Flow Capacity

Type: Table Slot
Units: Flow
Description: maximum possible flow rate through the distribution system
Information: The Maximum Flow Capacity can be based on either physical, legal, or environmental restrictions, etc.
I/O: Required input
Links: Not linkable

When Depletion Requested is input, Diversion Requested is computed by the following equation:

$$DiversionRequested = \mathrm{MIN}\left(MaximumFlowCapacity, \frac{DepletionRequested}{MinimumEfficiency}\right)$$

See the note **HERE** for information on how this method behaves in Rulebased Simulation.

27.1.1.4 Irrigation Requests

This user method uses irrigation data to determine Depletion Requested and/or Diversion Requested if they are not input. Depletion Requested is based on the area to be irrigated, evapotranspiration rate, and incidental rate loss. Diversion Requested is then calculated from the Depletion Requested and the Minimum Efficiency.

Slots Specific to this Method

☞ Evapotranspiration Rate

Type: Series Slot
Units: Length Vs Time
Description: rate of water loss due to evaporation and transpiration

Information:
I/O: Optional; required if Depletion Requested is not input.
Links: Not linkable

☞ INCIDENTAL LOSS RATE
Type: Table Slot
Units: FRACTION
Description: losses in the water distribution system
Information: Expressed as a fraction of the flow.
I/O: Optional; required if Depletion Requested is not input.
Links: Not linkable

☞ IRRIGATED AREA
Type: Series Slot
Units: AREA
Description: surface area of the land to be irrigated
Information:
I/O: Optional; required if Depletion Requested is not input.
Links: Not linkable

☞ MAXIMUM EFFICIENCY
Type: Table Slot
Units: FRACTION
Description: maximum amount of consumed flow per unit diverted flow
Information: Expressed as a fraction.
I/O: Required input
Links: Not linkable

☞ MAXIMUM FLOW CAPACITY
Type: Table Slot
Units: FLOW
Description: maximum possible flow rate through the distribution system
Information: The Maximum Flow Capacity can be based on either physical, legal, or environmental restrictions, etc.
I/O: Required input
Links: Not linkable

☞ MINIMUM DIVERSION REQUEST
Type: Table Slot
Units: FLOW
Description: minimum possible flow rate needed to meet depletion requests

Information: The Minimum Diversion Request can be based on either physical, legal, or environmental restrictions, etc. This is only used if the Depletion Request is non-zero.
I/O: Optional.
Links: Not linkable

☞ MINIMUM EFFICIENCY

Type: Series Slot
Units: FRACTION
Description: minimum amount of consumed flow per unit diverted flow
Information: Expressed as a fraction.
I/O: Required input
Links: Not linkable

If not input, the Depletion Requested is calculated by the following formula:

$$\text{Depletion Requested} = \text{Irrigated Area} \times \text{Evapotranspiration Rate} \times (1 + \text{Incidental Loss Rate})$$

Then, Diversion Requested (if it is not specified by the user) is calculated as follows:

$$\text{Diversion Requested} = \frac{\text{Depletion Requested}}{\text{Minimum Efficiency}}$$

After the Diversion Requested and Depletion Requested are initially calculated, Diversion Requested must be checked against the Maximum Flow Capacity. If the calculated value for Diversion Requested is less than the Maximum Flow Capacity, the Diversion Requested slot is set to the calculated value. If the calculated Diversion Requested is greater than the Max Flow Capacity, it is set to the Maximum Flow Capacity. (Note: user input overrides all calculated values. If the user specifies a Diversion Requested which is greater than the Maximum Flow Capacity, an error is posted and the run aborts.)

Diversion Requested must also be checked against the Minimum Diversion Request (if specified by user) If the calculated value for Diversion Requested is greater than the Minimum Diversion Request, the Diversion Requested slot is set to the calculated value. If the calculated Diversion Requested is less than the Minimum Diversion Request, it is set to the Minimum Diversion Request. (Note: user input overrides all calculated values. If the user specifies a Diversion Requested which is less than the Minimum Diversion Request (if specified), an error is posted and the run aborts.)

The Depletion Requested divided by the Diversion Requested (i. e. the Water User's efficiency) is compared to the Maximum Efficiency allowed on the Water User. If the calculated value is less than the Maximum Efficiency, Depletion Requested is unchanged. If the calculated efficiency (Depletion Requested/Diversion requested) is greater than the Maximum Efficiency (and Depletion Requested is not input by the user), Depletion Requested is reset to the value of Diversion Requested multiplied by the Maximum

Efficiency. If the calculated efficiency is greater than the Maximum Efficiency and Depletion Requested is input, an error is posted and the run aborts.

See the note **HERE** for information on how this method behaves in Rulebased Simulation.

27.1.1.5 Irrigation Requests with Soil Moisture

This method uses irrigation data to determine **Diversion** and **Depletion Requests** based on the irrigated acreage, evapotranspiration rate, and soil moisture. The Soil Moisture Demand and available soil moisture is also computed. This method is available for all Agg Diversion Site linking structure and for stand alone water users.

Note, you must also select the **Proportional Shortage with Soil Moisture** method, HERE (Section 27.1.4.4) or **Variable Efficiency with Soil Moisture** method, HERE (Section 27.1.4.6).

In addition, when this method is selected, the following dispatch methods are available based on the link structure:

- Stand alone Water User or Water User in a No Structure Agg Diversion Site:

  - Solve Stand alone given Depletion Requested

  - Solve Stand alone given Diversion

- Water User in a Sequential Agg Diversion Site:

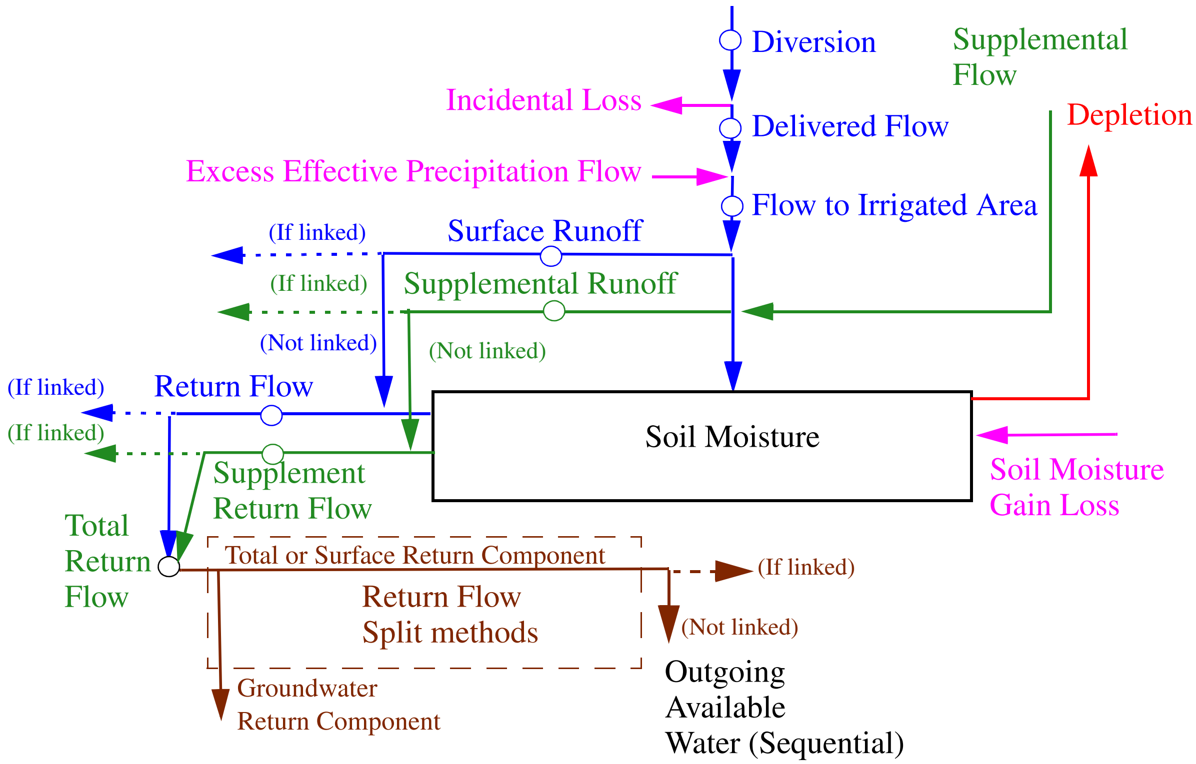
  - Solve Sequential given Depletion Requested

  - Solve Sequential given Diversion

- Water User in a Lumped Agg Diversion Site: This method cannot be selected. The Water user doesn’t dispatch in a Lumped Agg Diversion Site.

Below is a conceptual diagram of all of the components possible to model Soil Moisture on the Water User:

Items in blue are default flow values. Pink indicates optional gains/losses. Green is supplemental water (see the method HERE (Section 27.1.8.3) for more on supplemental flow). Brown indicates how the return flows can be split and/or routed. Note, not all values shown are separate slot values.



Slots Associated with this Method

☞ Evapotranspiration Rate

Type: Series Slot
Units: Length Vs Time
Description: rate of water loss due to evaporation and transpiration
Information:
I/O: Specified according to the selected method in the Irrigation Acreage and Evapotranspiration Rates category HERE (Section 27.1.2).
Links: Not linkable

☞ Incidental Loss Rate

Type: Table Slot
Units: Fraction
Description: losses in the water distribution system.
Information: Expressed as a fraction of the flow.
I/O: Optional; if not specified, zero is assumed.
Links: Not linkable

☞ Irrigated Area

Type: Series Slot

Units: AREA
Description: surface area of the land to be irrigated
Information:
I/O: Specified according to the selected method in the Irrigation Acreage and Evapotranspiration Rates category HERE (Section 27.1.2).
Links: Not linkable

☞ MAXIMUM FLOW CAPACITY

Type: Table Slot
Units: FLOW
Description: maximum possible flow rate through the distribution system
Information: The Maximum Flow Capacity can be based on either physical, legal, or environmental restrictions, etc.
I/O: Required input
Links: Not linkable

☞ MINIMUM DIVERSION REQUEST

Type: Table Slot
Units: FLOW
Description: minimum possible flow rate needed to meet depletion requests
Information: The Minimum Diversion Request can be based on either physical, legal, or environmental restrictions, etc. This is only used if the Depletion Request is non-zero.
I/O: Optional.
Links: Not linkable

☞ MINIMUM EFFICIENCY

Type: Series Slot
Units: FRACTION
Description: minimum amount of consumed flow per unit diverted flow
Information: Expressed as a fraction.
I/O: Required input
Links: Not linkable

☞ DIVERSION REQUEST FOR SOIL MOISTURE

Type: Series Slot
Units: FLOW
Description: The requested diversion explicitly to refill the soil moisture.
Information:
I/O: Output only
Links: Not linkable

☞ Diversion Request for Crops

Type: Series Slot
Units: Flow
Description: The requested diversion to meet crop depletion
Information:
I/O: Output only
Links: Not linkable

☞ Soil Moisture Future Timesteps

Type: Scalar
Units: No Units
Description: The number of timesteps past the current timestep for the calculations should be performed
Information: This must be an integer greater than 0.
I/O: If not specified, the value will be set to 1, which indicates that only the current timestep will be computed
Links: Not linkable

☞ Maximum Soil Moisture

Type: Scalar
Units: Length
Description: The depth of water that the soil can hold.
Information:
I/O: Required Input
Links: Not linkable

☞ Maximum Infiltration Rate

Type: Scalar
Units: Velocity (Length per Time)
Description: Maximum depth of water that can be absorbed by the soil
Information: A valid value must be specified or an error will occur at the beginning of run.
I/O: Required Input
Links: Not linkable

☞ Soil Moisture

Type: Series Slot
Units: Volume
Description: This slot tracks the volume of soil moisture water in the soil at each timestep in the run.
Information: For the initial timestep, this slot must be valid. If this slot is not input, it will be computed as the Maximum Soil Moisture times the Irrigated Area at the initial timestep. That is, the soil moisture starts full.
I/O: Typically Output for run timesteps.

Links: Not linkable

☞ Soil Moisture Demand Factor

Type: Series Slot with Periodic Input

Units: Fraction

Description: This slot is used to limit the soil moisture demand

Information:

I/O: Optional input as either a series or a periodic relationship. If not specified, it is assumed to be 1.

Links: Not linkable

☞ Soil Moisture Fill Efficiency

Type: Series Slot with Periodic Input

Units: Fraction

Description: The fraction of the flow applied for soil moisture refill that is not absorbed.

Information: For example, if you are only refilling the soil moisture, you may apply 100 cfs, but 5 cfs is not absorbed even though max infiltration indicated it could absorb 200cfs. The Soil Moisture Fill Efficiency would be 0.95.

I/O: Optional input, if not input, it is assumed to be 1.0.

Links: Not linkable

☞ Soil Moisture Demand

Type: Series Slot

Units: Flow

Description: The flow necessary to refill the soil moisture to capacity.

Information:

I/O: Output only

Links: Not linkable

☞ Soil Moisture Stress Coefficient

Type: Series Slot

Units: Fraction

Description: This coefficient represents a stress term that can be used to reduce the depletion request when in shortage.

Information:

I/O: Optional input or set by a rule. If not specified, the value is assumed to be 1.0, but no value is set on the slot. Since this method is executed at the beginning of the timestep, if you wish to set this slot with a rule, the rule must set it at the previous timestep.

Links: Not linkable

☞ Soil Moisture Gain Loss

Type: Series Slot

Units: VOLUME
Description: Miscellaneous gain or loss of soil moisture volume during the timestep.
Information: This slot represents and gains (positive) or loss (negative) from the soil moisture. It is used to track the soil moisture volume lost when the irrigated acreage is reduced from the previous timestep. In this case it is a computed output. If the irrigated acreage increases, you may wish to have the new acreage start with some soil moisture (perhaps due to precipitation). In this case, you can input a positive volume that will be added to the Soil Moisture.
I/O: Typically Output but can be Input
Links: Not Linkable

METHOD DETAILS

The **Irrigation Requests with Soil Moisture** method is executed in two parts, the first occurs at the beginning of each controller timestep and then the rest is executed from the dispatch method.

BEGINNING OF TIMESTEP:

Compute the Depletion Requested for all timesteps from the current timestep through the number of **Soil Moisture Future Timesteps**. This represents the crop demand for each of these timesteps.

$$\text{Depletion Requested} = \text{Irrigated Area} \times \text{Evapotranspiration Rate} \times \text{Soil Moisture Stress Coefficient}$$

AT THE START OF THE DISPATCH METHOD:

Recompute the Depletion Requested if any of the pieces have changed (by a rule).

$$\text{Depletion Requested} = \text{Irrigated Area} \times \text{Evapotranspiration Rate} \times \text{Soil Moisture Stress Coefficient}$$

Compute the maximum infiltration flow intermediate variable:

$$\text{maxInfiltrationFlow} = \text{Irrigated Area} \times \text{Max Infiltration Rate}$$

Compute the Soil Moisture Gain Loss (if not input or set by a rule)

If the current Irrigated Area is less than the Irrigated Area(t-1):

$$\text{Soil Moisture Gain Loss} = \frac{\text{Soil Moisture}(t-1)}{\text{Irrigated Area}(t-1)} \times \text{Irrigated Area} - \text{Soil Moisture}(t-1)$$

This loss will keep the new irrigated area at its same relatively fullness (at the start of the timestep), the lost water only comes from the area no longer irrigated. Note, Soil Moisture Gain Loss can also be input and will be used in the following equation.

Compute the flow necessary to refill the soil moisture back to the maximum (intermediate):

$$\text{flowToFillSoilMoisture} = \frac{\text{Maximum Soil Moisture} \times \text{Irrigated Area} - (\text{Soil Moisture}(t-1) + \text{Soil Moisture Gain Loss})}{\Delta t}$$

Compute the soil moisture demand, representing the physical constraints on refilling the soil moisture from diverted surface water:

$$\text{Soil Moisture Demand} = \frac{MIN(MAX(\text{maxInfiltrationFlow} - \text{Depletion Requested}, 0), \text{flowToFillSoilMoisture})}{\text{Soil Moisture Fill Efficiency}}$$

The Depletion Requested (crop demand) is subtracted from the *maxInfiltrationFlow* as all water applied to the fields must adhere to the max infiltration rate.

Compute the Diversion Request for Soil Moisture:

$$\text{Diversion Request for Soil Moisture} = \text{Soil Moisture Demand} \times \text{Soil Moisture Demand Factor}$$

Compute the Diversion Request for Crops:

$$\text{Diversion Request for Crops} = \frac{\text{Depletion Requested}}{\text{Minimum Efficiency}}$$

Compute the total Diversion Requested (if not input or set by a rule):

$$\text{Diversion Requested} = \frac{(\text{Diversion Request for Crops} + \text{Diversion Request for Soil Moisture})}{(1 - \text{Incidental Loss Rate})}$$

After the Diversion Requested and Depletion Requested are initially calculated, Diversion Requested is checked against the Maximum Flow Capacity. If the calculated value for Diversion Requested is less than the Maximum Flow Capacity, the Diversion Requested slot is set to the calculated value. If the calculated Diversion Requested is greater than the Max Flow Capacity, it is set to the Maximum Flow Capacity. (Note: user input overrides all calculated values. If the user specifies a Diversion Requested which is greater than the Maximum Flow Capacity, an error is posted and the run aborts.)

Diversion Requested must also be checked against the Minimum Diversion Request (if specified by user) If the calculated value for Diversion Requested is greater than the Minimum Diversion Request, the Diversion Requested slot is set to the calculated value. If the calculated Diversion Requested is less than the Minimum Diversion Request, it is set to the Minimum Diversion Request. (Note: user input overrides all calculated values. If the user specifies a Diversion Requested which is less than the Minimum Diversion Request (if specified), an error is posted and the run aborts.)

27.1.1.6 Head Gate Sprinkler Requests

The Head Gate Sprinkler Requests method uses agricultural data along with data on sprinkler evaporation to determine Depletion Requested and Diversion Requested. The method is very similar to the Irrigation Requests method except that a sprinkler evaporation rate is included in calculating Depletion Requested. This method is specific to the San Juan Basin and may not be physically valid in other situations. Use this method with caution.

Slots Specific to this Method

☞ Evapotranspiration Rate

Type: Series Slot
Units: LENGTH/TIME
Description: rate of water loss due to evaporation and transpiration
Information:
I/O: Optional; required if Depletion Requested is not input.
Links: Not linkable

☞ Incidental Loss Rate

Type: Table Slot
Units: Fraction
Description: losses in the water distribution system
Information: Expressed as a fraction of the flow.
I/O: Optional; required if Depletion Requested is not input.
Links: Not linkable

Links: Not linkable

☞ Irrigated Area

Type: Series Slot
Units: Area
Description: surface area of the land to be irrigated
Information:
I/O: Optional; required if Depletion Requested is not input.
Links: Not linkable

☞ Maximum Efficiency

Type: Table Slot
Units: Fraction
Description: maximum amount of consumed flow per unit diverted flow
Information: Expressed as a fraction.
I/O: Required input
Links: Not linkable

☞ Maximum Flow Capacity

Type: Table Slot
Units: Flow
Description: maximum possible flow rate through the distribution system
Information: The Maximum Capacity can be based on either physical, legal, or environmental, restrictions, etc.
I/O: Required input
Links: Not linkable

☞ Minimum Diversion Request

Type: Table Slot
Units: Flow
Description: minimum possible flow rate required to meet depletion requests
Information: The Minimum Diversion Request can be based on either physical, legal, or environmental restrictions, etc. This is only used if the Depletion Request is non-zero.
I/O: Optional.
Links: Not linkable

☞ Minimum Efficiency

Type: Series Slot
Units: Fraction
Description: minimum amount of consumed flow per unit diverted flow
Information:
I/O: Required input
Links: Not linkable

☞ Sprinkler Evaporation Rate

Type: Table Slot
Units: Fraction
Description: evaporative losses in the sprinkler mechanism
Information: Expressed as a fraction of the diversion.
I/O: Required input

Initially, the Depletion Requested is calculated by the following formula (if it is not input):

$$\text{Depletion Requested} = \text{Irrigated Area} \times \text{Evapotranspiration Rate} \times (1 + \text{Incidental Loss Rate})$$

Then, Diversion Requested (if it is not specified by the user) is calculated as follows:

$$\text{Diversion Requested} = \frac{\text{Depletion Requested}}{\text{Minimum Efficiency}}$$

After the Diversion Requested and Depletion Requested are initially calculated, the Diversion Requested must be checked against the Maximum Flow Capacity. If the calculated Diversion Requested is greater than the Max Flow Capacity, it is set to the Maximum Flow Capacity. If the calculated value for Diversion Requested is less than the Maximum Flow Capacity, the Diversion Requested slot is set with the calculated value. (Note: user input overrides all calculated values. If the user specifies a Diversion Requested which is greater than the Maximum Flow Capacity, an error is be posted and the run aborts).

Diversion Requested must also be checked against the Minimum Diversion Request (if specified by user) If the calculated value for Diversion Requested is greater than the Minimum Diversion Request, the Diversion Requested slot is set to the calculated value. If the calculated Diversion Requested is less than the Minimum Diversion Request, it is set to the Minimum Diversion Request. (Note: user input overrides all calculated values. If the user specifies a Diversion Requested which is less than the Minimum Diversion Request (if specified), an error is posted and the run aborts.)

The sprinkler evaporation is applied to the Depletion Requested after the Diversion Requested slot has been set. The initial Depletion Requested is adjusted as follows:

$$\text{Depletion Requested} = \text{Depletion Requested} + (\text{Diversion Requested} \times \text{Sprinkler Evaporation Rate})$$

The Depletion Requested divided by the Diversion Requested (i.e. the Water User's Efficiency) is compared to the Maximum Efficiency allowed on the water User. If the calculated Water User's efficiency is less than the Maximum Efficiency, the Depletion Requested Slot is set equal to the calculated value for Depletion Requested. If the calculated efficiency of the water user (Depletion Requested/Diversion requested) is greater than the Maximum Efficiency (and Depletion Requested is not input by the user), Depletion Requested is set to the product of the Diversion Requested and the Maximum Efficiency. If the calculated efficiency is greater than the Maximum Efficiency and Depletion Requested is input by the user, an error is posted and the run aborts.

See the note **HERE** for information on how this method behaves in Rulebased Simulation.

27.1.1.7 Population Requests

This user method uses population data to determine Depletion Requested and Diversion Requested if they are not input. Depletion Requested is based on the water user population, water use per individual, and incidental rate loss. Diversion Requested is then calculated from the Depletion Requested and the Minimum Efficiency.

Slots Specific to this Method

☞ Use Rate Per Individual

Type: Series Slot
Units: Flow
Description: water use as flow per individual
Information:
I/O: Optional; required if Depletion Requested is not input.
Links: Not linkable

☞ Incidental Loss Rate

Type: Table Slot
Units: Fraction
Description: losses in the water distribution system
Information: Expressed as a fraction of the flow.
I/O: Optional; required if Depletion Requested is not input.
Links: Not linkable

☞ Population

Type: Series Slot
Units: NoUnits
Description: the number of individuals in the water user population
Information:
I/O: Optional; required if Depletion Requested is not input.
Links: Not linkable

☞ Maximum Efficiency

Type: Table Slot
Units: Fraction
Description: maximum amount of consumed flow per unit diverted flow
Information: Expressed as a fraction.
I/O: Required input
Links: Not linkable

☞ Maximum Flow Capacity

Type: Table Slot
Units: Flow
Description: maximum possible flow rate through the distribution system
Information: The Maximum Flow Capacity can be based on either physical, legal, or environmental restrictions, etc.
I/O: Required input
Links: Not linkable

☞ Minimum Diversion Request

Type: Table Slot
Units: Flow
Description: minimum possible flow rate required to meet depletion requests
Information: The Minimum Diversion Request can be based on either physical, legal, or environmental restrictions, etc. This is only used if the Depletion Request is non-zero.
I/O: Optional.
Links: Not linkable

☞ Minimum Efficiency

Type: Series Slot
Units: Fraction
Description: minimum amount of consumed flow per unit diverted flow
Information: Expressed as a fraction.
I/O: Required input
Links: Not linkable

If not input, the Depletion Requested is calculated by the following formula:

$$\text{Depletion Requested} = \text{Population} \times \text{Use Rate Per Individual} \times (1 + \text{Incidental Loss Rate})$$

Then, Diversion Requested (if it is not specified by the user) is calculated as follows:

$$\text{Diversion Requested} = \frac{\text{Depletion Requested}}{\text{Minimum Efficiency}}$$

After the Diversion Requested and Depletion Requested are initially calculated, Diversion Requested must be checked against the Maximum Flow Capacity. If the calculated value for Diversion Requested is less than the Maximum Flow Capacity, the Diversion Requested slot is set to the calculated value. If the calculated Diversion Requested is greater than the Max Flow Capacity, it is set to the Maximum Flow Capacity. (Note: user input overrides all calculated values. If the user specifies a Diversion Requested which is greater than the Maximum Flow Capacity, an error is posted and the run aborts.)

Diversion Requested must also be checked against the Minimum Diversion Request (if specified by user) If the calculated value for Diversion Requested is greater than the Minimum Diversion Request, the Diversion Requested slot is set to the calculated value. If the calculated Diversion Requested is less than the Minimum Diversion Request, it is set to the Minimum Diversion Request. (Note: user input overrides all calculated values. If the user specifies a Diversion Requested which is less than the Minimum Diversion Request (if specified), an error is posted and the run aborts.)

The Depletion Requested divided by the Diversion Requested (i. e. the Water User's efficiency) is compared to the Maximum Efficiency allowed on the Water User. If the calculated value is less than the Maximum Efficiency, Depletion Requested is unchanged. If the calculated efficiency (Depletion Requested/Diversion requested) is greater than the Maximum Efficiency (and Depletion Requested is not input by the user), Depletion Requested is reset to the value of Diversion Requested multiplied by the Maximum Efficiency. If the calculated efficiency is greater than the Maximum Efficiency and Depletion Requested is input, an error is posted and the run aborts.

See the note **HERE** for information on how this method behaves in Rulebased Simulation.

27.1.1.8 Regional Requests

This user method uses regional data to determine Depletion Requested and Diversion Requested if they are not input. Depletion Requested is based on the regional water use, the fraction of the region that the water user represents, and incidental rate loss. Diversion Requested is then calculated from the Depletion Requested and the Minimum Efficiency.

Slots Specific to this Method

☞ **Regional Use Rate**

Type: Series Slot
Units: Flow

Description: water loss due to regional use
Information:
I/O: Optional; required if Depletion Requested is not input.
Links: Not linkable

☞ Incidental Loss Rate
Type: Table Slot
Units: Fraction
Description: losses in the water distribution system
Information: Expressed as a fraction of the flow.
I/O: Optional; required if Depletion Requested is not input.
Links: Not linkable

☞ Fraction of Region
Type: Series Slot
Units: FRaction
Description: fraction of the region using water at the regional rate
Information:
I/O: Optional; required if Depletion Requested is not input.
Links: Not linkable

☞ Maximum Efficiency
Type: Table Slot
Units: Fraction
Description: maximum amount of consumed flow per unit diverted flow
Information: Expressed as a fraction.
I/O: Required input
Links: Not linkable

☞ Maximum Flow Capacity
Type: Table Slot
Units: Flow
Description: maximum possible flow rate through the distribution system
Information: The Maximum Flow Capacity can be based on either physical, legal, or environmental restrictions, etc.
I/O: Required input
Links: Not linkable

☞ Minimum Diversion Request
Type: Table Slot
Units: Flow
Description: minimum possible flow rate required to meet depletion requests

Information: The Minimum Diversion Request can be based on either physical, legal, or environmental restrictions, etc. This is only used if the Depletion Request is non-zero.
I/O: Optional.
Links: Not linkable

☞ MINIMUM EFFICIENCY

Type: Series Slot
Units: FRACTION
Description: minimum amount of consumed flow per unit diverted flow
Information: Expressed as a fraction.
I/O: Required input
Links: Not linkable

If not input, the Depletion Requested is calculated by the following formula:

$$\text{Depletion Requested} = \text{Fraction of Region} \times \text{Regional Use Rate} \times (1 + \text{Incidental Loss Rate})$$

Then, Diversion Requested (if it is not specified by the user) is calculated as follows:

$$\text{Diversion Requested} = \frac{\text{Depletion Requested}}{\text{Minimum Efficiency}}$$

After the Diversion Requested and Depletion Requested are initially calculated, Diversion Requested must be checked against the Maximum Flow Capacity. If the calculated value for Diversion Requested is less than the Maximum Flow Capacity, the Diversion Requested slot is set to the calculated value. If the calculated Diversion Requested is greater than the Max Flow Capacity, it is set to the Maximum Flow Capacity. (Note: user input overrides all calculated values. If the user specifies a Diversion Requested which is greater than the Maximum Flow Capacity, an error is posted and the run aborts.)

Diversion Requested must also be checked against the Minimum Diversion Request (if specified by user) If the calculated value for Diversion Requested is greater than the Minimum Diversion Request, the Diversion Requested slot is set to the calculated value. If the calculated Diversion Requested is less than the Minimum Diversion Request, it is set to the Minimum Diversion Request. (Note: user input overrides all calculated values. If the user specifies a Diversion Requested which is less than the Minimum Diversion Request (if specified), an error is posted and the run aborts.)

The Depletion Requested divided by the Diversion Requested (i. e. the Water User's efficiency) is compared to the Maximum Efficiency allowed on the Water User. If the calculated value is less than the Maximum Efficiency, Depletion Requested is unchanged. If the calculated efficiency (Depletion Requested/Diversion requested) is greater than the Maximum Efficiency (and Depletion Requested is not input by the user), Depletion Requested is reset to the value of Diversion Requested multiplied by the Maximum

Efficiency. If the calculated efficiency is greater than the Maximum Efficiency and Depletion Requested is input, an error is posted and the run aborts.

See the note **HERE** for information on how this method behaves in Rulebased Simulation.

27.1.1.9 Periodic Diversion Request

This method is used to specify the Diversion Requested as a function of date/season using a periodic slot.

Slots Specific To This Method

☞ **Periodic Diversion Request**

Type: Periodic Slot
Units: FLOW
Description: The diversion requested as a function of date/season
Information: The Diversion Requested is computed at the beginning of each timestep using the current date to look up the value from this slot.
I/O: Required input
Links: Not linkable

The Periodic Diversion Request method executes at the beginning of the run. Each timestep is used to look up a value in the Periodic Diversion Request slot. The computed value is set in the Diversion Requested and Depletion Requested slots.

27.1.1.10 Reservoir Level Lookup

The Reservoir Level Lookup method is used to compute the Diversion Requested as a function of the date/season and the operating level of a specified reservoir.

Slots Specific To This Method

☞ **Level vs Diversion Request**

Type: Periodic Slot
Units: FLOW
Description: The diversion requested as a function of reservoir level and date/season
Information: Each column corresponds to a reservoir's operating level. For each operating level, the diversion requested is specified for each date range in the periodic slot.
I/O: Required input
Links: Not linkable

☞ **Diversion Request Reservoir**

Type: List Slot
Units: NONE

Description: The reservoir used to compute the diversion requested
Information: The previous timestep operating level of this reservoir is used to look up the diversion requested in the Level vs Diversion Request slot.
I/O: Required input
Links: Not linkable

☞ Forecast Period

Type: Scalar
Units: NONE
Description: The forecast period is a number of timesteps, including the current simulation timestep, that is used in the algorithms for calculating forecasted hydrology, regulation discharge and flood releases. This can be propagated from a computational subbasin of which the control point is a member.
Information: This slot must be input and greater than or equal to 1. If a Forecast Period is not necessary, input a value of 1.
I/O: Required Input
Links: Not Linkable

Note: If the specified reservoir is disabled and is set to Pass Inflows, no diversion request lookup is possible. See the note **HERE (USACE_SWD.pdf, Section 4.3.1)** on the behavior in this situation.

The Reservoir Level Lookup method executes at the beginning of each timestep. For each timestep (t) in the forecast period, the Diversion Request Reservoir's previous timestep's (current timestep - 1) operating level and the forecast timestep's date (t) will be used to look up the diversion request value in the Level vs Diversion Request slot. This value is set on the Diversion Requested and Depletion Requested slot. Because the method uses the same operating level (from the previous timestep) for all timesteps in the forecast period, it is an approximation but allows the object to solve throughout the forecast period.

27.1.1.11 Specify Scheduled Requests

The **Specify Scheduled Requests** method sets the **Diversion Requested** and **Depletion Requested** slots at the beginning of the run to the scheduled values. When the water user solves, it computes the amount of the depletion schedule that was cutback and the amount of the depletion schedule that couldn't be met. This allows you to track both an initial scheduled amount (as either a series or periodic relationship) and then reduce the schedule to the requests. Typically this reduction is performed using rules. The water user then computes the various depletion shortage computations.

Slots Specific To This Method

☞ Diversion Schedule

Type: Series Slot with Periodic Input
Units: FLOW
Description: The desired diversion before any cutback, curtailment or adjustments are made.
I/O: Input as either a Series or a Periodic relationship
Links: Not Linkable

☞ Depletion Schedule

Type: Series Slot with Periodic Input
Units: FLOW
Description: The desired diversion before any cutback, curtailment or adjustments are made.
I/O: Input as either a Series or a Periodic relationship
Links: Not Linkable

☞ Depletion Schedule Cutback

Type: Series Slot
Units: FLOW
Description: The difference between the Depletion Schedule and the Depletion Requested. This represents the amount that the schedule was adjusted to get to the request.
Information: Depletion Schedule Cutback[ ] = Depletion Schedule[ ] – Depletion Requested[ ]
I/O: Output Only
Links: Not linkable

☞ Depletion Schedule Shortage

Type: Series Slot
Units: FLOW
Description: The difference between the Depletion Schedule and the Depletion. A positive number indicates shortage, while a negative number indicates surplus.
Information: Depletion Schedule Shortage[ ] = Depletion Schedule[ ] – Depletion[ ]
I/O: Output only
Links: Not Linkable

The **Specify Scheduled Requests** method executes once at the beginning of the run for all timesteps in the run. The values in the **Diversion Schedule** and **Depletion Schedule** are copied to the **Diversion Requested** and **Depletion Requested** slots, respectively. Since these values are set at the beginning of the run, they are set when the controller is at priority 0 (in RBS) and the O flag. Therefore any rule executed later in the run can overwrite the values.

Shortage Computations:

When the **Specify Scheduled Requests** is selected, additional shortage terms are computed:

Stand Alone or part of a Sequential Agg Diversion Site: At the end of the dispatch method (HERE (Section 27.2.1.1) or HERE (Section 27.2.2.1)), the following are computed:

Depletion Schedule Cutback[ ] = Depletion Schedule[ ] – Depletion Requested[ ]

Depletion Schedule Shortage[ ] = Depletion Schedule[ ] – Depletion[ ]

Part of a Lumped Agg Diversion Site: When the aggregate dispatches (HERE (Section 3.2.4)), it computes the following on each water user:

Depletion Schedule Cutback[ ] = Depletion Schedule[ ] – Depletion Requested[ ]

Depletion Schedule Shortage[ ] = Depletion Schedule Cutback[ ] + Depletion Shortage[ ]

EXAMPLE:

Because of the complexity of these computations, following is a table with example numbers. The blue variables are part of this method. "NA" indicates it is not a slot on that object.

Variable	WU Stand Alone / Sequential	Lumped Agg Diversion Site with one water user element	
		Agg	WU element
Diversion Schedule	1000	NA	1000
Depletion Schedule	800	NA	800
Diversion Request	(rule) 900	900	(rule) 900
Depletion Request	(rule) 720	720	(rule) 720
Diversion (due to available)	850	850	NA
Depletion	680	680	NA
Depletion Shortage	40	40	40
Depletion Schedule Cutback	80	NA	80
Depletion Schedule Shortage	120	NA	120

27.1.2 Irrigation Acreage and Evapotranspiration Rates

This category is only available when the **Irrigation Requests** or **Irrigation Requests with Soil Moisture** user method is selected in the **Diversion and Depletion Request** category. The category provides methods that compute **Irrigated Area** and **Evapotranspiration Rate** slots.

27.1.2.1 Input Acreage and Rates

This is the default method. If this method is selected, Irrigated Area and Evapotranspiration Rate slots are optional inputs but required if Depletion Requested is not input.

27.1.2.2 Aggregate Acreage and Rates

This method calculates and sets the Irrigated Area and an area-weighted Evapotranspiration Rate for use by the Irrigation Requests user method in the Diversion and Depletion Request user category.

SLOTS SPECIFIC TO THIS METHOD

☞ **IRRIGATED AREA BY CROP**

Type: AggSeries Slot
Units: AREA
Description: The irrigated area of each crop
Information: Each user-specified column represents a different irrigated crop acreage. These columns correspond to the crop evapotranspiration rates specified in the Evapotranspiration Rate by Crop AggSeries slot. The column labels can be changed to represent the given crop.
I/O: Input only
Links: Not linkable

☞ **EVAPOTRANSPIRATION RATE BY CROP**

Type: AggSeries Slot
Units: VELOCITY
Description: The evapotranspiration rate of each crop
Information: Each user-specified column represents a different irrigated crop evapotranspiration rate as volume per area per time. These columns correspond to the crops specified in the Irrigated Area by Crop AggSeries slot. The column labels can be changed to represent the given crop.
I/O: Input only
Links: Not linkable

The method will calculate and set the Irrigated Area and an area-weighted Evapotranspiration Rate in the Irrigation Requests user method in the Diversion and

Depletion Request user category on the Water User object. Solution logic for the Aggregate Acreage and Rates method proceeds as follows:

Given the Crop-specific Irrigation Area and the Crop-specific Evapotranspiration Rate slots with columns 0 to n:

$$\text{Irrigated Area} = \sum_{i=0}^{n} Area_i$$

If Irrigated Area is 0, meaning no irrigation is occurring:

$$\text{Evapotranspiration Rate} = 0$$

Otherwise:

$$\text{Evapotranspiration Rate} = \frac{\sum_{i=0}^{n} Area_i \times Rate_i}{\text{Irrigated Area}}$$

27.1.3 Limit Diversion

This category is only available when either **Periodic Diversion Request** or **Reservoir Level Lookup** is selected in the **Diversion and Depletion Request** category. The information added by these methods is used by a pre-defined RPL function called **ComputeReservoirDiversions**, **HERE (RPLPredefinedFunctions.pdf, Section 20)**, to determine reservoir diversions.

27.1.3.1 None

There are no slots or calculation associated with this method.

27.1.3.2 Limit by Reservoir Level

This method adds the Demand Reservoir slot. This slot is used to specify the reservoir that is receiving the diverted water (this water will pass through the Water User object and into the reservoir via the Return Flow slot). If this method is selected, the RPL function ComputeReservoirDiversions, **HERE (RPLPredefinedFunctions.pdf, Section 20)**, that computes the Supply From Reservoirs slot values will not compute a diversion if the Demand Reservoir is in the flood pool or if its level is higher than the supply reservoir.

SLOTS SPECIFIC TO THIS METHOD

☞ **DEMAND RESERVOIR**

Type: List Slot
Units: NONE
Description: The demand reservoir - the reservoir receiving the diverted water
Information: This slot is used by a pre-defined RPL function (ComputeReservoirDiversions, **HERE (RPLPredefinedFunctions.pdf, Section 20)**) that computes multiple reservoir diversions to meet multiple demands. As part of the RPL function logic, if this reservoir is in the flood pool, or if its operating level is higher than the supply reservoir, then a diversion is not made (the RPL function computes a value of zero for the associated subslot in the Supply From Reservoirs slot).
I/O: Required input
Links: Not linkable

27.1.4 Return Flow

The Return Flow category is dependent on the user selecting either the No Structure or Sequential Structure linking structure or the Water User being stand-alone. The user methods within the Return Flow category specify how the return flow will be calculated for each Water User. The user has the option of linking the return flow back to the diverted object or elsewhere along the system.

These methods are executed after the Diversion slot has been set.

27.1.4.1 None

None is an invalid method and an error will be posted and the run aborted when the user tries to run a model with **None** selected in the **Return Flow** category. (Note: this is the default method for the **Return Flow** category.)

27.1.4.2 Fractional Return Flow

The Fractional Return Flow method calculates the return flow based on a user input fraction. The input fraction (Fractional Return Flow) is representative of the amount of flow that is not consumed by the Water User.

SLOTS SPECIFIC TO THIS METHOD

☞ **FRACTION RETURN FLOW**

Type: Series Slot
Units: FRACTION
Description: fraction of the diverted flow that is not consumed by the Water User
Information:
I/O: Required Input
Links: Not linkable

☞ **RETURN FLOW**

Type: Series Slot
Units: FLOW
Description: amount of diverted water not consumed by Water User
Information:
I/O: Output only
Links: May be linked to the Return Flow slot on any object, or the Inflow slot on a Groundwater Storage object.

This user method performs very simple calculations. The Return Flow is calculated as shown below:

$$\text{Return Flow} = \text{Diversion} \times \text{Fractional Return Flow}$$

The Depletion (or amount of water consumed by the water user) is calculated as shown below:

$$\text{Depletion} = \text{Diversion} - \text{Return Flow}$$

The above calculations are only performed if the Fractional Return Flow is known. If the Fractional Return Flow is not known, a **RiverWare**™ error is posted and the run is aborted.

27.1.4.3 Proportional Shortage

The Proportional Shortage Method computes the Return Flow based on the ratio of Diversion to Diversion Requested and the Depletion Requested.

Slots Specific to this Method

☞ Return Flow

Type: Series Slot
Units: FLOW
Description: amount of diverted water not consumed by the Water User
Information:
I/O: Output only
Links: May be linked to the Return Flow slot on any object, or the Inflow slot on a Groundwater Storage object.

This method uses a set of simple calculations and logical expressions to determine the Return Flow.

The delivered fraction (a local variable) is determined as follows. If the Diversion Requested is equal to 0.0, the delivered fraction is set equal to 1.0. If the Diversion Requested is not equal to 0.0, the delivered fraction is set by the following equation:

$$\text{delivered fraction} = \frac{\text{Diversion}}{\text{Diversion Requested}}$$

After the delivered fraction is determined, the following equation is used to determine the Depletion:

$$\text{Depletion} = \text{Depletion Requested} \times \text{delivered fraction}$$

The Return Flow is then calculated as follows:

$$\text{Return Flow} = \text{Diversion} - \text{Depletion}$$

27.1.4.4 Proportional Shortage with Soil Moisture

This method uses the diverted water to first meet **Depletion Requested**. If that is not sufficient, then **Soil Moisture** can be used to meet the Depletion Requested. This method also tracks the volume of water stored in the soil moisture.

This method is only available for stand alone water users and water user elements that are part of a No Structure or Sequential structure Agg Diversion site. Also, you must select the **Irrigation Requests with Soil Moisture** method HERE (Section 27.1.1.5).

See a diagram of the Soil Moisture slots HERE (Section 27.1.1.5).

SLOTS ASSOCIATED WITH THIS METHOD.

☞ **AVAILABLE SOIL MOISTURE FRACTION**

Type: Series
Units: FRACTION
Description: The fraction (decimal or percentage) of the moisture in the soil that is available for use by the crops
Information:
I/O: Output Only
Links: Not linkable

☞ **CROP USAGE**

Type: Series
Units: FLOW
Description: The amount of water used by the crops. This includes depletion of delivered water and soil moisture usage, but not supplemental flow.
Information:
I/O: Output only
Links: Not Linkable

☞ **CROP SHORTAGE**

Type: Series
Units: FLOW
Description: The amount of crop demand that is not met by either diversions or soil moisture.
Information:
I/O: Output only
Links: Not Linkable

☞ **CONSUMPTIVE USE FROM DELIVERED FLOW**

Type: Series Slot
Units: FLOW
Description: The portion of the consumed water that came from the Diversion.

Information: This is identical to Depletion.
I/O: Output Only
Links: Not Linkable

☞ Consumptive Use from Soil Moisture
Type: Series Slot
Units: Flow
Description: The portion of the consumed water that came from the soil moisture
Information:
I/O: Output Only
Type: Not Linkable

☞ Delivered Flow
Type: Series Slot
Units: Flow
Description: The portion of the diverted flow that is delivered to the fields and is available for crops or to refill the soil moisture.
Information: This is computed as the Diversion minus the Incidental Loss.
I/O: Output only
Links: Not Linkable

☞ Diversion Request for Crop
Type: Series Slot
Units: Flow
Description: The requested diversion to meet crop depletion
Information: This is computed by the Irrigation Requests with Soil Moisture method HERE (Section 27.1.1.5).
I/O: Output only
Links: Not linkable

☞ Efficiency including Soil Moisture
Type: Series Slot
Units: Fraction
Description: Efficiency calculation that includes the flow to the soil moisture.
Information: It shows how water is used beneficially since water stored in soil moisture will be consumed later by the crop. This slot is only used for informational purposes and is not used in further calculations.
I/O: Output only

☞ Incidental Loss
Type: Series Slot
Units: Flow
Description: The portion of the Diversion that is lost due to the incidental loss rate.

Information:
I/O: Output only
Links: Not Linkable

☞ Incidental Loss Rate
Type: Table Slot
Units: Fraction
Description: Losses in the water distribution system.
Information: Expressed as a fraction of the flow.
I/O: Optional; if not specified, zero is assumed.
Links: Not linkable

☞ Irrigated Area
Type: Series Slot
Units: Area
Description: Surface area of the land to be irrigated
Information:
I/O: Specified according to the selected method in the Irrigation Acreage and Evapotranspiration Rates category HERE (Section 27.1.2).
Links: Not linkable

☞ Maximum Soil Moisture
Type: Scalar
Units: Length
Description: The depth of water that the soil can hold.
Information:
I/O: Required Input
Links: Not linkable

☞ Maximum Infiltration Rate
Type: Scalar
Units: Velocity (Length per Time)
Description: Maximum depth of water that can be absorbed by the soil
Information: A valid value must be specified or an error will occur at the beginning of run.
I/O: Required Input
Links: Not linkable

☞ Return Flow
Type: Series Slot
Units: Flow
Description: Amount of diverted water not consumed by the Water User
Information:
I/O: Output only

Links: May be linked to flow slots on any object

☞ SOIL MOISTURE

Type: Series Slot
Units: VOLUME
Description: This slot tracks the volume of soil moisture water in the soil at each timestep in the run.
Information: For the initial timestep, this slot must be valid. If this slot is not input, it will be computed as the Maximum Soil Moisture times the Irrigated Area at the initial timestep. That is, the soil moisture starts full.
I/O: Typically Output for run timesteps.
Links: Not linkable

☞ SOIL MOISTURE DEMAND

Type: Series Slot
Units: FLOW
Description: The flow necessary to refill the soil moisture to capacity.
Information:
I/O: Output only
Links: Not linkable

☞ SOIL MOISTURE FILL EFFICIENCY

Type: Series Slot with Periodic Input
Units: FRACTION
Description: The fraction of the flow applied for soil moisture refill that is not absorbed.
Information: For example, if you are only refilling the soil moisture, you may apply 100 cfs, but 5 cfs is not absorbed even though max infiltration indicated it could absorb 200cfs. The Soil Moisture Fill Efficiency would be 0.95.
I/O: Optional input. If not specified, it is assumed to be zero
Links: Not linkable

☞ SOIL MOISTURE FLOW

Type: Series
Units: FLOW
Description: The flow into or out of the soil moisture.
Information: A positive number indicates flow into the soil moisture storage. A negative number indicates the soil moisture is being used to meet crop demands.
I/O: Output only
Links: Not linkable

☞ SURFACE RUNOFF

Type: Series
Units: FLOW

Description: The water that runs off the field because the maximum infiltration rate has been exceeded. This water never enters the soil.
Information: If the Surface Runoff slot **is** linked to a slot on another object it is not included in the total Return Flow. If the Surface Runoff **is not** linked, the Surface Runoff is included as part of the total Return Flow.
I/O: Output only
Links: Linkable to another object's Return Flow slot

☞ EXCESS EFFECTIVE PRECIPITATION
Type: Series with Periodic Input
Units: VELOCITY (LENGTH/TIME)
Description: "Excess" means effective precipitation that typically cannot be used by the crops. Normally, effective precipitation is subtracted from the crop demand to compute a net Depletion Requested. However, in some instances the effective precipitation is greater than what the crops could use within a given time step. Therefore, a value for excess effective precipitation would likely be input only when input Depletion Requested is zero (because the crop demand would have been met by the "non-excess" portion of the effective precipitation). Excess Effective Precipitation goes towards filling the soil moisture column until the available capacity is reached. Any remaining excess effective precipitation that cannot be stored in the soil moisture column is added to return flow
Information: This value is multiplied by the irrigated acreage to compute a flow.
I/O: Input or set by a rule. It is assumed to be zero if not input.
Links: Not Linkable

☞ EXCESS EFFECTIVE PRECIPITATION FLOW
Type: Series
Units: FLOW
Description: The flow rate that represents the excess effective precipitation
Information: **Excess Effective Precipitation Flow** = **Excess Effective Precipitation** * **Irrigated Area**
I/O: Output only
Links: Not linkable

The **Proportional Shortage with Soil Moisture** Return Flow method does the following:

First, compute how much of diversion is lost due to conveyance or "incidental" loss.

$$\text{Incidental Loss} = \text{Diversion} \times \text{Incidental Loss Rate}$$

Calculate how much of the diverted flow is actually delivered:

$$\text{Delivered Flow} = \text{Diversion} - \text{Incidental Loss}$$

Calculate the Excess Effective Precipitation flow:

$$\text{Excess Effective Precipitation Flow} = \text{Irrigated Area} \times \text{Excess Effective Precipitation}$$

Calculate the temporary variable *flowToIrrigatedArea*. This is the flow applied to the field:

$$flowToIrrigatedArea = \text{Delivered Flow} + \text{Excess Effective Precipitation Flow}$$

Compute the maximum infiltration flow intermediate variable:

$$\text{maxInfiltrationFlow} = \text{Irrigated Area} \times \text{Max Infiltration Rate}$$

Compute the flow the is above the maxInfiltrationFlow. This water does not absorb and runs directly off the field's surface.

$$\text{Surface Runoff} = MAX(flowToIrrigatedArea - \text{maxInfiltrationFlow}, 0cms)$$

Compute the flow that infiltrates to the soil:

$$flowInfiltrated = flowToIrrigatedArea - \text{Surface Runoff}$$

Compute the Depletion. Depletion is limited by the *maxInfiltrationFlow* as it cannot use more than is infiltrated.

$$\text{Depletion} = MIN(\text{maxInfiltrationFlow}, flowToIrrigatedArea \times \text{Minimum Efficiency, Depletion Requested})$$

Calculate the initial return flow intermediate value. This will occur for any diversion even when storing soil moisture.

$$throughSoilReturnInitial = MIN(flowInfiltrated, \text{Diversion Request for Crops}) - \text{Depletion}$$

Calculate the initial crop shortage. This is the amount of the depletion requested that is not met by diverted water:

$$c\text{ropShortageInitial} = \text{Depletion Requested} - \text{Depletion}$$

Calculate Soil Moisture Flow:

IF (cropShortageInitial > 0)

$$\text{Soil Moisture Flow} = -(MIN(\text{cropShortageInitial, Soil Moisture[t-1]}/(\Delta t)))$$

This is negative indicating water is moving from the soil moisture to the crops.

ELSE

$$\text{Soil Moisture Flow} = \text{Soil Moisture Fill Efficiency} \times MIN(flowInfiltrated - \text{Depletion} - throughSoilReturnInitial, \text{Soil Moisture Demand})$$

Extra water may have been diverted (if the demand factor was 0, or Diversion was specified) or there may be excess effective precipitation. The above equation uses the Soil Moisture Demand, so it can refill the soil moisture, even if the demand factor is zero. When water is applied for soil moisture, there is some that is returned, according to the Soil Moisture Fill Efficiency.

END IF

Calculate final Return Flow. If the Surface Runoff slot is linked to another object it is not included in the total Return Flow:

$$\text{Return Flow} = flowToIrrigatedArea - \text{Surface Runoff} - \text{Depletion} - MAX(\text{Soil Moisture Flow,0})$$

If this Surface Runoff is not linked, the Surface Runoff is included as part of the total Return Flow:

$$\text{Return Flow} = flowToIrrigatedArea - \text{Depletion} - MAX(\text{Soil Moisture Flow,0})$$

Calculate Soil Moisture and Available Fraction:

$$\text{Soil Moisture[t]} = \text{Soil Moisture}[t-1] + \text{Soil Moisture Gain Loss} + \text{Soil Moisture Flow} \times \Delta t$$

$$\text{Available Soil Moisture Fraction} = \frac{\text{Soil Moisture}}{\text{Maximum Soil Moisture} \times \text{Irrigated Area}}$$

Diversion Shortage:

$$\text{Diversion Shortage} = \text{Diversion Requested} - Diversion$$

Depletion Shortage:

$$\text{Depletion Shortage} = \text{Depletion Requested} - \text{Depletion}$$

Compute total crop usage and shortage:

$$\text{Crop Usage} = \text{Depletion} - MIN(0,\text{Soil Moisture Flow})$$

$$\text{Crop Shortage} = \text{Depletion Requested} - \text{Crop Usage}$$

Finally compute summary variables:

$$\text{Consumptive Use from Delivered Flow} = \text{Depletion}$$

$$\text{Consumptive Use from Soil Moisture} = -MIN(\text{Soil Moisture Flow, 0})$$

$$\text{Efficiency including Soil Moisture} = \frac{\text{Depletion} + MAX(\text{Soil Moisture Flow,0})}{\text{Delivered Flow} + \text{Excess Effective Precipitation Flow}}$$

27.1.4.5 Variable Efficiency

The Variable Efficiency method computes the Return Flow based on an efficiency driven approach. This method (or Variable Efficiency with Soil Moisture) must be selected if the user wishes to use the SW GW Efficiency Split method in the Return Flow Split category.

Slots Specific to this Method

☞ Efficiency

Type: Series Slot
Units: Fraction
Description: amount of water depleted per unit water diverted for a given timestep
Information: Expressed as a fraction.
I/O: Output only
Links: Not linkable

☞ Maximum Efficiency

Type: Table Slot
Units: Fraction
Description: maximum amount of water depleted per unit water diverted by the Water User
Information: Expressed as a fraction.
I/O: Required Input
Links: Not linkable

☞ Return Flow

Type: Series Slot
Units: Flow
Description: amount of diverted water not consumed by the Water User
Information:
I/O: Output only
Links: May be linked to the Return Flow slot on any object, or the Inflow slot on a Groundwater Storage object.

The engineering algorithm associated with this method is given below.

The first step in this algorithm is to check that the Maximum Efficiency is greater than the ratio of Depletion Requested to Diversion Requested. If it is not, a **RiverWare**™ error is posted and the run is aborted. If Depletion Requested is not known, it is assumed to be the same as Diversion Requested.

If the Maximum Efficiency is greater than the ratio of Depletion Requested to Diversion Requested, the Efficiency is calculated as follows:

$$\text{Efficiency} = \frac{\text{Depletion Requested}}{\text{Diversion}}$$

If the calculated Efficiency is greater than the Maximum Efficiency, the Efficiency is reset to value of the Maximum Efficiency.

The Depletion is calculated as follows:

$$\text{Depletion} = \text{Efficiency} \times \text{Diversion}$$

After the Depletion has been calculated, it is used to calculate the Return Flow.

$$\text{Return Flow} = \text{Diversion} - \text{Depletion}$$

(Note: this method must be selected to use the Split Return Efficiency method in the Return Flow Split category.)

27.1.4.6 Variable Efficiency with Soil Moisture

The Variable Efficiency with Soil Moisture method computes the Return Flow based on an efficiency driven approach. Diverted water first meets **Depletion Requested**. If that is not sufficient, then **Soil Moisture** can be used to meet the **Depletion Requested**. This method also tracks the volume of water stored in the soil moisture.

This method is only available for stand alone water users and water user elements that are part of a No Structure or Sequential structure Agg Diversion site. Also, you must select the **Irrigation Requests with Soil Moisture** method **HERE (Section 27.1.1.5)**.

See also a diagram of the use of Soil Moisture slots **HERE (Section 27.1.1.5)**.

Slots Specific to this Method

☞ Efficiency

Type: Series Slot
Units: Fraction
Description: amount of water depleted per unit water diverted for a given timestep
Information: Expressed as a fraction.
I/O: Output only
Links: Not linkable

☞ Efficiency including Soil Moisture

Type: Series Slot
Units: Fraction
Description: Efficiency calculation that includes the flow to the soil moisture.
Information: It shows how water is used beneficially since water stored in soil moisture will be consumed later by the crop. This slot is only used for informational purposes and is not used in further calculations.
I/O: Output only
Links: Not Linkable

☞ Available Soil Moisture Fraction

Type: Series
Units: Fraction
Description: The fraction (decimal or percentage) of the moisture in the soil that is available for use by the crops
Information:
I/O: Output Only
Links: Not linkable

☞ Crop Usage

Type: Series
Units: Flow
Description: The amount of water used by the crops. This includes depletion of delivered water and soil moisture usage, but not supplemental flow.
Information:
I/O: Output only
Links: Not Linkable

☞ Crop Shortage

Type: Series
Units: Flow
Description: The amount of crop demand that is not met by either diversions or soil moisture.
Information:
I/O: Output only
Links: Not Linkable

☞ Consumptive Use from Delivered Flow

Type: Series Slot
Units: Flow
Description: The portion of the consumed water that came from the Diversion.
Information: This is identical to Depletion.
I/O: Output Only
Links: Not Linkable

☞ Consumptive Use from Soil Moisture

Type: Series Slot
Units: Flow
Description: The portion of the consumed water that came from the soil moisture
Information:
I/O: Output Only
Links: Not Linkable

☞ Delivered Flow

Type: Series Slot
Units: Flow
Description: The portion of the diverted flow that is delivered to the fields and is available for crops or to refill the soil moisture.
Information: This is computed as the Diversion minus the Incidental Loss.
I/O: Output only
Links: Not Linkable

☞ Diversion Request for Crop

Type: Series Slot
Units: Flow
Description: The requested diversion to meet crop depletion
Information: This is computed by the Irrigation Requests with Soil Moisture method HERE (Section 27.1.1.5).
I/O: Output only
Links: Not linkable

☞ Incidental Loss

Type: Series Slot
Units: Flow
Description: The portion of the Diversion that is lost due to the incidental loss rate.
Information:
I/O: Output only
Links: Not Linkable

☞ Incidental Loss Rate

Type: Table Slot
Units: Fraction
Description: Losses in the water distribution system.
Information: Expressed as a fraction of the flow.
I/O: Optional; if not specified, zero is assumed.
Links: Not linkable

☞ Irrigated Area

Type: Series Slot
Units: Area
Description: Surface area of the land to be irrigated
Information:
I/O: Specified according to the selected method in the Irrigation Acreage and Evapotranspiration Rates category HERE (Section 27.1.2).
Links: Not linkable

☞ Maximum Soil Moisture

Type: Scalar
Units: LENGTH
Description: The depth of water that the soil can hold.
Information:
I/O: Required Input
Links: Not linkable

☞ Maximum Infiltration Rate

Type: Scalar
Units: VELOCITY (LENGTH PER TIME)
Description: Maximum depth of water that can be absorbed by the soil
Information: A valid value must be specified or an error will occur at the beginning of run.
I/O: Required Input
Links: Not linkable

☞ Return Flow

Type: Series Slot
Units: FLOW
Description: Amount of diverted water not consumed by the Water User
Information:
I/O: Output only
Links: May be linked to flow slots on any object

☞ Soil Moisture

Type: Series Slot
Units: VOLUME
Description: This slot tracks the volume of soil moisture water in the soil at each timestep in the run.
Information: For the initial timestep, this slot must be valid. If this slot is not input, it will be computed as the Maximum Soil Moisture times the Irrigated Area at the initial timestep. That is, the soil moisture starts full.
I/O: Typically Output for run timesteps.
Links: Not linkable

☞ Soil Moisture Demand

Type: Series Slot
Units: FLOW
Description: The flow necessary to refill the soil moisture to capacity.
Information:
I/O: Output only
Links: Not linkable

☞ Soil Moisture Fill Efficiency

Type: Series Slot with Periodic Input
Units: FRACTION
Description: The fraction of the flow applied for soil moisture refill that is not absorbed.
Information: For example, if you are only refilling the soil moisture, you may apply 100 cfs, but 5 cfs is not absorbed even though max infiltration indicated it could absorb 200cfs. The Soil Moisture Fill Efficiency would be 0.95.
I/O: Optional input. If not specified, it is assumed to be zero
Links: Not linkable

☞ Soil Moisture Flow

Type: Series
Units: FLOW
Description: The flow into or out of the soil moisture.
Information: A positive number indicates flow into the soil moisture storage. A negative number indicates the soil moisture is being used to meet crop demands.
I/O: Output only
Links: Not linkable

☞ Surface Runoff

Type: Series
Units: FLOW
Description: The water that runs off the field because the maximum infiltration rate has been exceeded. This water never enters the soil.
Information: If the Surface Runoff slot **is** linked to a slot on another object it is not included in the total Return Flow. If the Surface Runoff **is not** linked, the Surface Runoff is included as part of the total Return Flow.
I/O: Output only
Links: Linkable to another object’s Return Flow slot

☞ Excess Effective Precipitation

Type: Series
Units: VELOCITY (LENGTH/TIME)
Description: “Excess” means effective precipitation that typically cannot be used by the crops. Normally, effective precipitation is subtracted from the crop demand to compute a net Depletion Requested. However, in some instances the effective precipitation is greater than what the crops could use within a given time step. Therefore, a value for excess effective precipitation would likely be input only when input Depletion Requested is zero (because the crop demand would have been met by the “non-excess” portion of the effective precipitation). Excess Effective Precipitation goes towards filling the soil moisture column until the available capacity is reached. Any remaining

excess effective precipitation that cannot be stored in the soil moisture column is added to return flow

Information: This value is multiplied by the irrigated acreage to compute a flow.
I/O: Input or set by a rule. It is assumed to be zero if not input.
Links: Not Linkable

☞ Excess Effective Precipitation Flow

Type: Series
Units: Flow
Description: The flow rate that represents the excess effective precipitation
Information: **Excess Effective Precipitation Flow** = **Excess Effective Precipitation** * **Irrigated Area**
I/O: Output only
Links: Not linkable

The **Variable Efficiency with Soil Moisture** Return Flow method does the following:

First, compute how much of diversion is lost due to conveyance or “incidental” loss.

$$\text{Incidental Loss} = \text{Diversion} \times \text{Incidental Loss Rate}$$

Calculate how much of the diverted flow is delivered:

$$\text{Delivered Flow} = \text{Diversion} - \text{Incidental Loss}$$

Calculate the Excess Effective Precipitation flow:

$$\text{Excess Effective Precipitation Flow} = \text{Irrigated Area} \times \text{Excess Effective Precipitation}$$

Calculate the temporary variable *flowToIrrigatedArea*. This is the flow applied to the field:

$$flowToIrrigatedArea = \text{Delivered Flow} + \text{Excess Effective Precipitation Flow}$$

Compute the maximum infiltration flow intermediate variable:

$$\text{maxInfiltrationFlow} = \text{Irrigated Area} \times \text{Max Infiltration Rate}$$

Compute the flow the is above the maxInfiltrationFlow. This water does not absorb and runs directly off the field’s surface.

$$\text{Surface Runoff} = MAX(flowToIrrigatedArea - \text{maxInfiltrationFlow}, 0cms)$$

Compute the flow that infiltrates to the soil:

$$flowInfiltrated = flowToIrrigatedArea - \text{Surface Runoff}$$

The Efficiency is calculated as follows:

$$\text{Efficiency} = MIN\left(\frac{MIN(\text{Depletion Requested, maxInfiltrationFlow})}{MIN(flowToIrrigatedArea, \text{Diversion Request for Crops})}, \text{Maximum Efficiency}\right)$$

Thus, the Efficiency should always be between the Min and Max Efficiency, unless limited by max infiltration. If limited by max infiltration, the efficiency can be less than the minimum.

Compute the Depletion. Depletion is limited by the *maxinfiltrationFlow* as it cannot use more than is infiltrated.

$$\text{Depletion} = MIN(\text{ maxInfiltrationFlow}, flowToIrrigatedArea \times \text{Efficiency, Depletion Requested})$$

Calculate the initial return flow intermediate value. This will occur for any diversion even when storing soil moisture. This water enters the soil but is not consumed.

$$throughSoilReturnInitial = MIN(flowInfiltrated, \text{Diversion Request for Crops}) - \text{Depletion}$$

Calculate the initial crop shortage. This is the amount of the depletion requested that is not met by diverted water:

$$\text{cropShortageInitial} = \text{Depletion Requested} - \text{Depletion}$$

Calculate Soil Moisture Flow:

IF (cropShortageInitial > 0)

$$\text{Soil Moisture Flow} = -(MIN(\text{cropShortageInitial, Soil Moisture[t-1]}/(\Delta t)))$$

This is negative indicating water is moving from the soil moisture to the crops.

ELSE

$$\text{Soil Moisture Flow} = \text{Soil Moisture Fill Efficiency} \times MIN(flowInfiltrated - \text{Depletion} - throughSoilReturnInitial, \text{Soil Moisture Demand})$$

Extra water may have been diverted (if the demand factor was 0, or Diversion was specified). The above equation uses the Soil Moisture Demand, so it can refill the soil moisture, even if the demand factor is zero. When water is applied for soil moisture, there is some that is returned, according to the Soil Moisture Fill Efficiency.

END IF

Calculate final Return Flow. If the Surface Runoff slot is linked to another object it is not included in the total Return Flow:

$$\text{Return Flow} = flowToIrrigatedArea - \text{Surface Runoff} - \text{Depletion} - MAX(\text{Soil Moisture Flow}, 0)$$

If this Surface Runoff is not linked, the Surface Runoff is included as part of the total Return Flow:

$$\text{Return Flow} = flowToIrrigatedArea - \text{Depletion} - MAX(\text{Soil Moisture Flow}, 0)$$

Calculate Soil Moisture and Available Fraction:

$$\text{Soil Moisture[t]} = \text{Soil Moisture}[t-1] + \text{Soil Moisture Gain Loss} + \text{Soil Moisture Flow} \times \Delta t$$

$$\text{Available Soil Moisture Fraction} = \frac{\text{Soil Moisture}}{\text{Maximum Soil Moisture} \times \text{Irrigated Area}}$$

Diversion Shortage:

$$\text{Diversion Shortage} = \text{Diversion Requested} - Diversion$$

Depletion Shortage:

$$\text{Depletion Shortage} = \text{Depletion Requested} - \text{Depletion}$$

Compute total crop usage and shortage:

$$\text{Crop Usage} = \text{Depletion} - MIN(0, \text{Soil Moisture Flow})$$

$$\text{Crop Shortage} = \text{Depletion Requested} - \text{Crop Usage}$$

Finally compute summary variables:

$$\text{Consumptive Use from Delivered Flow} = \text{Depletion}$$

$$\text{Consumptive Use from Soil Moisture} = -MIN(\text{Soil Moisture Flow}, 0)$$

$$\text{Efficiency including Soil Moisture} = \frac{\text{Depletion} + MAX(\text{Soil Moisture Flow}, 0)}{\text{Delivered Flow} + \text{Excess Effective Precipitation Flow}}$$

27.1.4.7 Pump Back Return Flow

The Pump Back Return Flow Method allows a percentage of return flow from a water user to be returned to the diverted water for reuse. This method requires that the user selects Impulse Response as a routing method and None in the Return Flow Split category.

SLOTS SPECIFIC TO THIS METHOD

☞ **RETURN FLOW**

Type: Series Slot
Units: FLOW
Description: Amount of diverted water that is not consumed by the user.

I/O: Output Only
Links: May be linked to the Return Flow slot on any object, or the Inflow slot on a Groundwater Storage object.

☞ Incidental Depletions

Type: Scalar Slot
Units: FRACTION
Description: Single value that describes the percentage of diverted water lost to seepage or leakage.
Information: Decimal Value
I/O: Required input
Links: Not Linkable

☞ Irrigated Area

Type: Series Slot
Units: AREA
Description: Area of land that is irrigated by the user.
I/O: Required Input
Links: Not Linkable

☞ Consumptive irrigation requirement

Type: Series Slot
Units: VELOCITY
Description: Rate of water consumption per unit area of irrigated land.
I/O: Required Input
Links: Not Linkable

☞ Application Efficiency

Type: Scalar Slot
Units: FRACTION
Description: The fraction of applied water that is consumed by the crops
I/O: Required input
Links: Not linkable

☞ Pump back flow

Type: Series Slot
Units: FLOW
Description: Amount of Routed Return Flow that is pumped back into use.
Information: This slot may be set by a rule.
I/O: Optional; if not input, Pump Back Flows are not considered
Links: Not Linked

☞ PUMP BACK DEPLETION

Type: Scalar Slot
Units: FLOW
Description: Table slot describing the percentage of Pump Back Flow that is depleted by the user.
I/O: Required Input
Links: Not Linked

☞ PUMP BACK RETURN

Type: Series Slot
Units: FLOW
Description: Amount of water (from Pump Back Flow) that is available for return after Pump Back Depletions are considered.
I/O: Output Only
Links: Not Linked

☞ PROJECT EFFICIENCY

Type: Scalar Slot
Units: FRACTION
Information: Expressed as a fraction
I/O: Required Input
Links: Not Linked

☞ NON APPLIED WATER

Type: Series Slot
Units: FLOW
Description: Unused diverted water
I/O: Input or Set by a Rule
Links: May be Linked

The routing lag coefficient is checked first to be sure that the initial value has been set to zero. This prevents an iterative loop from occurring between Return Flow, Pump Back Return, and Routed Return Flow. The Project Efficiency is also checked to make sure it is not zero. This prevents division by zero when calculating the Non Applied Water. Non Applied Water can then be input or set by a rule.

If Non Applied Water is a value less than zero, it is set equal to zero. Otherwise the value calculated above is used.

Return flow is then computed as:

$$ReturnFlow(t) = MAX((1 - ApplicationEfficiency) \times Diversion(t), Diversion(t) - [(1 + IncidentalDepletion) \times IrrigatedArea \times ConsumptiveIrrigationRequirement])$$

Next, Depletion is calculated as:

$$Depletion = Diversion - ReturnFlow - NonAppliedWater$$

After Return Flow and Depletion are calculated, the method checks to see if there are pump back flows. If so, return flow from above is calculated with the following two equations:

$$PumpBackReturn(t) = (1 - PumpBackDepletion) \times PumpBackFlow(t)$$

$$ReturnFlow(t) = ReturnFlow(t) + PumpBackReturn(t)$$

The Routed Return Flow is adjusted as follows:

$$RoutedReturnFlow(t) = RoutedReturnFlow(t) - PumpBackFlow(t)$$

27.1.5 Fraction Return Flow Input

The Fraction Return Flow Input category is dependent on the user selecting the Fraction Return Flow method in the Return Flow category. Three methods allow the user to specify how the Fractional Return Flow (series slot) is provided data: Input Fraction, Zero Fraction, and Periodic Fraction.

The methods in this category will be called at the beginning of the run to set the Fractional Return Flow slot at each timestep.

27.1.5.1 Input Fraction

This default method requires that the Fractional Return Flow slot be input at each timestep. RiverWare will abort with an error if the Fractional Return Flow is not input.

27.1.5.2 Zero Fraction

At the beginning of the run, this method sets the Fractional Return Flow to 0.0 for each timestep. Hence the Return Flow will always be zero. No slots are specific to this method. RiverWare will abort with an error if the Fractional Return Flow is input.

27.1.5.3 Periodic Fraction

This method allows the user to specify the return flow fraction using the Periodic Fraction slot. The user can input a value of 1.0 to simulate the case where all water is returned. At the beginning of the run, each timestep is used to look-up or interpolate (depending on slot configuration) the value from the Periodic Fraction and set it on the Fractional Return Flow slot. If this value is not between 0.0 and 1.0 on a given timestep, RiverWare will abort with an error. RiverWare will abort with an error if the Fractional Return Flow is input.

SLOTS SPECIFIC TO THIS METHOD

☞ **PERIODIC FRACTION**

Type: Periodic
Units: TIME VS FRACTION
Description: Fraction of the diverted flow that is not consumed by the Water User.
I/O: Input Only
Links: Not Linkable

27.1.6 Return Flow Split

The Return Flow Split category is dependent selecting either the No Structure or Sequential linking structure on the Agg Diversion Site or the Water User being stand-alone. This category contains methods which allow the user to “split” the return flow into surface water and groundwater components or multiple components. The None, SW GW Fractional Split and MultiReturn Fractional Split methods are available regardless of which Return Flow method is chosen. The SW GW Efficiency Split method is only available when the Variable Efficiency or Variable Efficiency with Soil Moisture methods are selected in the Return Flow category. If the user has selected Supplement Diversion in the Conjunctive Use category, the Total Return Flow, which is the sum of the Return Flow and the Supplement Return Flow, will be split into components. Otherwise, it is just the Return Flow that is split.

For each method below, the final step describes how the return flow split and the linked slots affect a sequential Agg Diversion Site. In each method, it lists the part of the return flow that contributes to the Outgoing Available Water based on linking of return flow slots.

27.1.6.1 None

This method performs no computations.

There is no split into surface and groundwater components. The return flow is treated as a single entity. This is the default method for the Return Flow Split category.

When part of a sequential Agg Diversion Site the Return Flow Split determines method determines how much contributes to the next water user elementn. If using a Conjunctive Use method:

- If the Total Return Flow is linked, then the Total Return Flow is not included in the outgoing available flow.
- If the Total Return Flow is not linked, Return Flow is included in the outgoing available flow.

If not using a Conjunctive Use method:

- If the Return Flow is linked, then the Return Flow is not included in the outgoing available flow.
- If the Return Flow is not linked, Return Flow is included in the outgoing available flow.

27.1.6.2 SW GW Fractional Split

The SW GW Fractional Split method splits the Return Flow (plus Supplement Return Flow if applicable) into surface water and groundwater components based on a user input fraction.

SLOTS SPECIFIC TO THIS METHOD

☞ **FRACTION GW RETURN FLOW**

Type: Series Slot
Units: FRACTION

Description: amount of groundwater return flow per unit total return flow
Information: Expressed as a fraction.
I/O: Optional; Required when GW Return Flow and Surface Return Flow are not input.
Links: Usually not linked.

☞ GW Return Flow

Type: Series Slot
Units: Flow
Description: amount of return flow to groundwater
Information:
I/O: Optional; Required when Fraction GW Return Flow and Surface Return Flow are not input.
Links: Inflow slot on a Groundwater Storage object, series slot on Data Object.

☞ Surface Return Flow

Type: Series Slot
Units: Flow
Description: amount of return flow to surface water
Information:
I/O: Optional; Required when Fraction GW Return Flow and GW Return Flow are not input.
Links: Return Flow slot on any object, series slot on Data Object.

If the total return flow (sum of Return Flow and Supplement Return Flow) is equal to 0.0, both the GW Return Flow and the Surface Return Flow are set equal to 0.0. If the total return flow is not equal to 0.0 and Surface Return Flow is input (assuming both Fraction GW Return Flow and GW Return Flow are not input), GW Return is calculated by the following equation:

$$\text{GW Return Flow} = \text{total return flow} - \text{Surface Return Flow}$$

Fraction GW Return Flow is then calculated as:

$$\text{Fraction GW Return Flow} = \frac{\text{GW Return Flow}}{\text{total return flow}}$$

If GW Return Flow is input and both Fraction GW Return Flow and Surface Return Flow are not input, Surface Return Flow is calculated as:

$$\text{Surface Return Flow} = \text{total return flow} - \text{GW Return Flow}$$

Fraction GW Return Flow is then calculated as:

$$\text{Fraction GW Return Flow} = \frac{\text{GW Return Flow}}{\text{total return flow}}$$

If both Surface Return Flow and GW Return Flow are NOT input, the following steps are taken: If the total return flow is not equal to 0.0 and the Fraction GW Return Flow is input (and both GW Return Flow and Surface Return Flow are not input), the following equations are used to calculate the GW Return Flow and the Surface Return Flow. The GW Return Flow is shown below:

$$\text{GW Return Flow} = \text{Fraction GW Return Flow} \times \text{total return flow}$$

The Surface Return Flow is the difference between the total return flow and the GW Return Flow, as shown in the equation below.

$$\text{Surface Return Flow} = \text{total return flow} - \text{GW Return Flow}$$

When part of a sequential Agg Diversion Site, the Surface Return Flow is included in the outgoing available flow when Surface Return Flow is not linked. If Surface Return Flow is linked, it goes elsewhere and is not available for downstream elements.

27.1.6.3 SW GW Efficiency Split

The SW GW Efficiency Split method is available only when the Variable Efficiency or Variable Efficiency with Soil Moisture methods are selected in the Return Flow category. The Efficiency value calculated from the Variable Efficiency method is used to determine the portion of return flow that is groundwater and the portion that is surface water.

Slots Specific to this Method

☞ GW Return Flow

Type: Series Slot
Units: Flow
Description: the amount of return flow to groundwater
Information:
I/O: Optional.
Links: Inflow slot on a Groundwater Storage object, series slot on a Data Object.

☞ GW Split Adjust Factor

Type: Table Slot
Units: Fraction
Description: amount of groundwater return flow per unit total return flow
Information:
I/O: Optional; Set to 1.0 if not input.
Links: Not linkable.

☞ Surface Return Flow

Type: Series Slot
Units: Flow

Description: amount of return flow to surface water
Information:
I/O: Optional.
Links: Return Flow slot on any object, series slot on Data Object.

If Surface Return Flow is input and GW Return Flow is not input, GW Return Flow is calculated by the following Equation:

$$\text{GW Return Flow} = \text{total return flow} - \text{Surface Return Flow}$$

If GW Return Flow is input and Surface Return Flow is not input, Surface Return Flow is calculated by the following equation:

$$\text{Surface Return Flow} = \text{total return flow} - \text{GW Return Flow}$$

If both Surface Return Flow and GW Return Flow are NOT input, the following steps are taken: If the total return flow (sum of the Return Flow and Supplement Return Flow) is equal to 0.0, both the GW Return Flow and the Surface Return Flow are set equal to 0.0. If the total return flow is not equal to 0.0, the GW Return Flow and the Surface Return Flow are calculated by the following equations.

The GW Return Flow is calculated using this equation:

$$\text{GW Return Flow} = \text{GW Split Adjust Factor} \times \frac{\text{Efficiency}}{\text{Maximum Efficiency}} \times \text{total return flow}$$

The surface Return Flow is the difference between the total return flow and the GW Return Flow as shown by the equation below.

$$\text{Surface Return Flow} = \text{total return flow} - \text{GW Return Flow}$$

When part of a sequential Agg Diversion Site, the Surface Return Flow is included in the outgoing available flow when Surface Return Flow is not linked. If Surface Return Flow is linked, it goes elsewhere and is not available for downstream elements.

27.1.6.4 Irrigated Area GW Return Rate

The Irrigated Area GW Return Rate method partitions Return Flow into GW Return Flow using irrigated area and groundwater return rate, with any remaining return flow assigned to Surface Return Flow.

Slots Specific to this Method

☞ Irrigated Area

Type: Series Slot
Units: AREA
Description: The irrigated area contributing to return flow.

Information: Also used by the Irrigated Requests method, if active
I/O: Input or Output.
Links: Not linkable.

☞ Groundwater Return Rate

Type: Series Slot
Units: VELOCITY
Description: The rate that water percolates into the ground.
Information: This represents the volume returning to groundwater per area of irrigated land per unit of time
I/O: Input only.
Links: Not linkable.

☞ GW Return Flow

Type: Series Slot
Units: FLOW
Description: The portion of the total Return Flow that percolates into the groundwater.
Information:
I/O: Output only.
Links: Linkable to the Inflow from Surface Water series slot on the connected Groundwater object.

☞ Surface Return Flow

Type: Series Slot
Units: FLOW
Description: The portion of the total return flow that remains surface water
Information:
I/O: Output only.
Links: Return Flow slot on any object, series slot on Data Object.

If Diversion Requested is 0, meaning no irrigation is occurring:

$$\text{Actual Irrigated Area} = 0$$

Otherwise:

$$\text{Actual Irrigated Area} = \text{Irrigated Area} \times \left(\frac{\text{Diversion}}{\text{Diversion Requested}}\right)$$

Using the user-input Groundwater Return Rate, which is constrained to be greater than or equal to 0:

$$\text{GW Return Flow} = \text{Actual Irrigated Area} \times \text{Groundwater Return Rate}$$

With the calculated total return flow (Return Flow plus Supplement Return Flow, where applicable) constraining GW Return Flow, if necessary:

$$\text{Surface Return Flow} = \text{total return flow} - \text{GW Return Flow}$$

When part of a sequential Agg Diversion Site, the Surface Return Flow is included in the outgoing available flow when Surface Return Flow is not linked. If Surface Return Flow is linked, it goes elsewhere and is not available for downstream elements.

27.1.6.5 Multi Return Fractional Split

This method is used to split the return flow into multiple return flow destinations. The user specifies the portion of the total return flow that goes to each destination.

Slots Specific to this Method

☞ **Return Flow Proportion**

Type: Table Slot
Units: Fraction
Description: a table giving the proportion of the total return flow that goes to each Returned Flows subslot
Information: The values in this slot should add up to 1.0. For example, if there are two columns and the total return flow is split evenly, 0.5 would go in one column and 0.5 would go in the other column. The table will have the same number of columns as the number of return flows. The columns will be added and named (by RiverWare) whenever a link is added to the Returned Flows multislot. A decimal number will be input for each column representing each return flow slot.
I/O: Required input
Links: Not linkable

☞ **Returned Flows**

Type: No Compute Multi Slot
Units: Flow
Description: the multiple (or split) return flows based on the Return Flow Proportion table
Information: The subslots of this multislot contain the multiple return flows. A subslot is created when the user links this slot to a destination object.
I/O: Output
Links: Linked to the return flow destination objects. If this slot is linked, the Return Flow slot should not be linked.

The multiple return flows would then be calculated as:

$$\text{Returned Flows (i)} = \text{Return Flow Proportion (i)} \times \text{total return flow}$$

where i is the column of the Return Flow Proportion table; and total return flow is the sum of Return Flow and Supplement Return Flow, where applicable.

The first column on the Returned Flows multislot is the multislot itself and is not one of the subslots. It is set as the sum of all the subslot values. Therefore it should be equal to the total return flow.

Each of the multiple return flows will be represented by a subslot in the Returned Flows multislot. A subslot is created by linking the Returned Flows slot to the destination object. Whenever a new subslot is created, a new column will be added to the Return Flow Proportion slot. The column will have the same name as the subslot so that the user can keep track of which subslot goes with which column.

When part of a sequential Agg Diversion Site, the Returned Flows are not included in the outgoing available flow. The Returned Flows are all linked, so flow goes elsewhere and is not available for downstream elements.

27.1.7 Return Flow Routing

The methods in the Return Flow Routing category are used to route the return flow on the Water User object. The methods available in this category are dependent upon the method selected in the Return Flow Split category.

27.1.7.1 None

This is the default method. It performs no calculations and instantiates no new slots. The user should use this method if the Return Flow is not routed.

27.1.7.2 Impulse Response

This method is available if the user has selected None in the Return Flow Split category. In this method the Return Flow is routed based on the impulse response method of routing. It sets values only on the dispatching timestep, i.e. current object timestep.

Slots Specific to this Method

☞ **Number of Coeffs**

Type: Scalar Slot
Units: None
Description: the integer number of lag coefficients to be used in the method.
I/O: Required input
Links: Not linkable

☞ **Lag Coefficients**

Type: Table Slot
Units: None
Description: the impulse response lag coefficients
Information: The same number of coefficients must be input as the value in the Number of Coeffs slot. The input will be in rows.
I/O: Required input
Links: Not linkable

☞ **Routed Return Flow**

Type: Series Slot
Units: Flow
Description: the routed return flow
Information: The return flows will be set in the Return Flow slot and then routed to this slot.
I/O: Output
Links: Should be linked to the destination object. If this slot is linked the Return Flow slot should not be linked.

The Routed Return Flow will be calculated as:

$$\text{Routed Return Flow}_t = C_0\text{Return Flow}_t + C_1\text{Return Flow}_{t-1} + \dots$$
$$+ C_{ncoeff-2}\text{Return Flow}_{t-(ncoeff-2)} + C_{ncoeff-1}\text{Return Flow}_{t-(ncoeff-1)}$$

In this equation, ReturnFlow_t is either the Return Flow slot or the Total Return Flow, when supplemental diversions are used.

When part of a sequential Agg Diversion Site, the Routed Return Flow is included in the outgoing available flow when Routed Return Flow and Return Flow are not linked. If either of these slots is linked, flow goes elsewhere and is not available for downstream elements.

27.1.7.3 Step Response

This method is available if the user has selected None in the Return Flow Split category. In this method the Return Flow is routed based on the step response method of routing. It sets values at timesteps forward of the dispatching timestep, i.e. future timesteps.

Slots Specific to this Method

☞ Number of Coeffs

Type: Scalar Slot
Units: None
Description: the integer number of lag coefficients to be used in the method.
I/O: Required input
Links: Not linkable

☞ Lag Coefficients

Type: Table Slot
Units: None
Description: the impulse response lag coefficients
Information: The same number of coefficients must be input as the value in the Number of Coeffs slot. The input will be in rows.
I/O: Required input
Links: Not linkable

☞ Routed Return Flow

Type: Series Slot
Units: Flow
Description: the routed return flow
Information: The return flows will be set in the Return Flow slot and then routed to this slot.
I/O: Output
Links: Should be linked to the destination object. If this slot is linked the Return Flow slot should not be linked.

☞ **Future Routing Timesteps**

Type: Scalar
Units: None
Description: The number of timesteps in the future (i.e. past the timestep at which the object is dispatching) at which Routed Return Flow will be computed and set.
Information: This slot must be input and greater than or equal to 1. If you do not want it to set future timesteps, input a value of 1.
I/O: Required Input
Information: Not Linkable

The calculations for this method are similar to the Impulse Response method, but this method sets values out into the future:

nFutureTimesteps = Min (Future Routing Timesteps, Number of Coeffs)

For each timestep k from (t to (t + nFutureTimesteps - 1))

$$\text{Routed Return Flow}_k = C_0\text{Return Flow}_k + C_1\text{Return Flow}_{k-1} + \ldots + C_{nCoeff-1}\text{Return Flow}_{k-ncoeff-1} + C_{nCoeff}\text{Return Flow}_{k-nCoeff}$$

End for

In this equation, ReturnFlow_k is either the Return Flow slot or the Total Return Flow, when supplemental diversions are used.

Thus, the method computes and sets Routed Return Flows forward nFutureTimesteps.

In the situation where the method is looking for a Return Flow for a timestep that is actually past the dispatching timestep and this value is not valid, this Return Flow is assumed to be zero. If the Return Flow is not valid for a previous timestep, the calculation will exit the method and post a warning message.

If the Routed Return Flow is input or specified by a rule, that value is used and the routing is not performed.

When part of a sequential Agg Diversion Site, the Routed Return Flow is included in the outgoing available flow when Routed Return Flow and Return Flow are not linked. If either of these slots is linked, flow goes elsewhere and is not available for downstream elements

27.1.7.4 SW GW Impulse Response

This method is available when either SW GW Fractional Split or SW GW Efficiency Split is selected in the Return Flow Split category.

Slots Specific to this Method

☞ **Number of SW GW Coeffs**

Type: Table Slot
Units: None

Description: a two column table slot giving the number of lag coefficients to be used in the routing method
Information: The first column will give the number of coefficients for the surface return flow and the second column will give the number of coefficients for the groundwater return flow.
I/O: Required input
Links: Not linkable

☞ SW GW Lag Coefficients

Type: Table Slot
Units: NONE
Description: the impulse response lag coefficients for both surface water and groundwater return flows
Information: The first column will contain the lag coefficients for the surface water return flow. There must be the same number of coefficients in this column as the value in the first column of the Number of Coeffs slot. The second column will contain the lag coefficients for the groundwater return flow. There must be the same number of coefficients in this column as the value in the second column of the Number of Coeffs slot.
I/O: Required input
Links: Not linkable

☞ SW Routed Return

Type: Series Slot
Units: FLOW
Description: the routed surface water return flow
I/O: This slot is computed by the routing method.
I/O: Output
Links: Should be linked to the destination object. If this slot is linked, the Surface Return Flow slot should not be linked.

☞ GW Routed Return

Type: Series Slot
Units: FLOW
Description: the routed groundwater return flow
Information: This slot is computed by the routing method.
I/O: Output
Links: Should be linked to the destination object. If this slot is linked, the GW Return Flow slot should not be linked.

The return flows will be calculated as:

$$\text{SW Routed Return}_t = C_0 \text{ Surface Return Flow}_t + C_1 \text{ Surface Return Flow}_{t-1} + \ldots + C_{ncoeff-1} \text{ Surface Return Flow}_{t-ncoeff-1} + C_{ncoeff} \text{ Surface Return Flow}_{t-ncoeff}$$

$$\text{GW Routed Return}_t = C_0 \text{ GW Return Flow}_t + C_1 \text{ GW Return Flow}_{t-1} + \ldots + C_{ncoeff-1} \text{ GW Return Flow}_{t-ncoeff-1} + C_{ncoeff} \text{ GW Return Flow}_{t-ncoeff}$$

When part of a sequential Agg Diversion Site, the SW Routed Return Flow is included in the outgoing available flow when SW Routed Return Flow and Return Flow are not linked. If either of these slots is linked, flow goes elsewhere and is not available for downstream elements

27.1.7.5 Multi Split Impulse Response

This method is available when Multi Return Fractional Split method is selected in the Return Flow Split category. Each split return flow is routed (using different parameters) by the impulse response method of routing. This method sets values only at the timestep at which the water user is dispatching.

Slots Specific to this Method

☞ Number of MR Coeffs

Type: Scalar Slot
Units: None
Description: the integer number of lag coefficients to be used in the method.
Information: A column will be added automatically for each of the split return flows. The name of the column is used to match it with the appropriate subslot in the Returned Flows slot. Each column corresponds to the number of lag coefficients to be used for that particular split return flow.
I/O: Required input
Links: Not linkable

☞ Multi Return Lag Coeffs

Type: Table Slot
Units: None
Description: the impulse response lag coefficients
Information: The same number of coefficients must be input in each column as the value in the corresponding column in the Number of MR Coeffs slot. The columns are added and named automatically as a new subslot is added to the Returned Flows slot.
I/O: Required input
Links: Not linkable

☞ PreRouted Return Flows

Type: Agg Series Slot
Units: Flow
Description: the return flows prior to routing

Information: A column will be added automatically for every subslot in the Returned Flows multislot. The name of the column is used to match it with the appropriate subslot in the Returned Flows slot.
I/O: Output (Can be input prior to the Start date to set initial values for impulse response routing).
Links: Not linkable

Subslots in the Returned Flows slot are created when a link is made from this slot to the destination object. When a subslot is created, a corresponding column in the Number of MR Coeffs, Multi Return Lag Coeffs and PreRouted Return Flows slots is added automatically. The new columns have the same name as the subslot so you know which column goes with which subslot.

The impulse response calculations are performed for every return flow split; the method is nested in a loop that executes for every column in the Number of MR Coeffs slot. The Returned Flows are calculated as:

$$\begin{aligned}\text{Returned Flows(i)}_t &= C(i)_0\ \text{PreRouted Return Flows(i)}_t \\ &+ C(i)_1 \text{PreRouted Return Flows(i)}_{t-1} + \ldots. \\ &+ C(i)_{ncoeff-1} \text{PreRouted Return Flows(i)}_{t-ncoeff-1} \\ &+ C(i)_{ncoeff} \text{PreRouted Return Flows(i)}_{t-ncoeff}\end{aligned}$$

The first column on the Returned Flows multislot is the multislot itself and is not one of the subslots. It is set as the sum of all the subslot values.

27.1.7.6 Multi Split Step Response

This method is available when Multi Return Fractional Split method is selected in the Return Flow Split category. Each split return flow is routed (using different parameters) by the step response method of routing. It sets values at timesteps forward of the dispatching timestep, i.e. future timesteps.

Slots Specific to this Method

☞ Number of MR Coeffs

Type: Scalar Slot
Units: NONE
Description: the integer number of lag coefficients to be used in the method.
Information: A column will be added automatically for each of the split return flows. The name of the column is used to match it with the appropriate subslot in the Returned Flows slot. Each column corresponds to the number of lag coefficients to be used for that particular split return flow.
I/O: Required input
Links: Not linkable

☞ MULTI RETURN LAG COEFFS

Type: Table Slot
Units: NONE
Description: the impulse response lag coefficients
Information: The same number of coefficients must be input in each column as the value in the corresponding column in the Number of MR Coeffs slot. The columns are added and named automatically as a new subslot is added to the Returned Flows slot.
I/O: Required input
Links: Not linkable

☞ PREROUTED RETURN FLOWS

Type: Agg Series Slot
Units: FLOW
Description: the return flows prior to routing
Information: A column will be added automatically for every subslot in the Returned Flows multislot. The name of the column is used to match it with the appropriate subslot in the Returned Flows slot.
I/O: Output or input prior to the Start date to set initial values for step response routing).
Links: Not linkable

☞ FUTURE ROUTING TIMESTEPS

Type: Scalar
Units: NONE
Description: The number of timesteps in the future (i.e. past the timestep at which the object is dispatching) at which Routed Return Flow will be computed and set.
Information: This slot must be an input integer and greater than or equal to 1. If you do not want it to set future timesteps, input a value of 1.
I/O: Required Input
Information: Not Linkable

The first column on the Returned Flows multislot is the multislot itself and is not one of the subslots. It is set as the sum of all the subslot values.

Subslots in the Returned Flows slot are created when a link is made from this slot to the destination object. When a subslot is created, a corresponding column in the Number of MR Coeffs, Multi Return Lag Coeffs and PreRouted Return Flows slots will be added automatically. The new columns will have the same name as the subslot so the user knows which column goes with which subslot.

The step response calculations are performed for every return flow split, the method is nested in a loop that executes for every column (i) in the Number of MR Coeffs slot.

The calculations for this method are similar to the Multi Split Impulse Response method, but this method sets values into the future.

nFutureTimesteps = Min (Future Routing Timesteps, Number of Coeffs)

For each timestep k from (t to (t + nFutureTimesteps - 1))

$$\text{Returned Flows(i)}_k = C(i)_0 \text{ PreRouted Return Flows(i)}_k + C(i)_1 \text{ PreRouted Return Flows(i)}_{k-1} + \ldots + C(i)_{ncoeff-1} \text{ PreRouted Return Flows(i)}_{k-ncoeff-1} + C(i)_{ncoeff} \text{ PreRouted Return Flows(i)}_{k-nCoeff}$$

End for

Thus, the method computes and sets Routed Return Flows forward nFutureTimesteps.

When the method is looking for a PreRouted Return Flow for a timestep that is past the dispatching timestep and the value is not valid, the PreRouted Return Flow is assumed to be zero. If the PreRouted Return Flow is not valid for a previous timestep, the calculation will exit the method and post a warning message. No Returned Flows will be set.

If the slot linked to the Returned Flows slot is input or set by a rule, the computed Returned Flow is not computed for that timestep as the value is already specified.

27.1.8 Conjunctive Use

The Conjunctive Use methods are used to model diversions from a different source (E.g. groundwater) to supplement surface water diversions.

27.1.8.1 None

This is the default user method. It performs no calculations and has no slots specifically associated with it.

27.1.8.2 Supplement Diversion

This method functions differently for the three linking structures. There is also a different set of slots for each of the linking structures.

The slot list for the default (No Structure) linking structure or for a stand-alone Water User is given below.

Slots Specific to this Method

☞ **Available Supplemental Water**

Type: Series Slot
Units: Flow
Description: represents the amount of water available to supplement surface water diversion
Information:
I/O: Required input; however, it can by set through a link.
Links: Usually linked to the Available For Pumping slot on a Groundwater Storage object.

☞ **Consumption**

Type: Series Slot
Units: Flow
Description: total amount of water consumed/used by the Water User
Information:
I/O: Output only
Links: Usually not linked

☞ **Consumption Shortage**

Type: Series Slot
Units: Flow
Description: the total water shortage experienced by the crops
Information: This represents the shortage in terms of the water requested for consumption.
I/O: Output only
Links: Usually not linked

☞ Supplemental Diversion

Type: Series Slot
Units: Flow
Description: the amount of water diverted from groundwater for supplemental use
Information:
I/O: Output only
Links: Should be linked to the Pumped Flow slot on the Groundwater Storage object.

☞ Supplement Return Flow

Type: Series Slot
Units: Flow
Description: the amount of return flow from the supplemental water
Information: Calculated by the same method that calculates return flow from surface diversions.
I/O: Output only
Links: May be linked to the Return Flow slot on any object or the Inflow slot on a Groundwater Storage object.

☞ Supplemental Use Delivery

Type: Series Slot
Units: Flow
Description: amount of supplemental water received by the Water User
Information: For the No Structure case, this is always equal to the Supplemental Diversion.
I/O: Output only
Links: Not linked for the No Structure case.

☞ Supplemental Use Request

Type: Series Slot
Units: Flow
Description: the amount of supplemental water requested
Information:
I/O: Optional; If not input, it is set to the lower value of Diversion Shortage or Maximum Supplement Request.
Links: Not linkable

☞ Total Applied Water

Type: Series Slot
Units: Flow
Description: total amount of water available for consumption/use by the water user
Information:
I/O: Output only
Links: Not linkable

☞ Total Return Flow

Type: Series Slot
Units: Flow
Description: the sum of the Return Flow and the Supplement Return Flow
Information: This value is split into surface and groundwater components if a method is selected in the Return Flow Split category.
I/O: Output only
Links: May be linked to the Return Flow slot on any object.

The slot list for the sequential linking structure is given below:

Slots Specific to this Method

☞ Consumption

Type: Series Slot
Units: Flow
Description: total amount of water consumed/used by the Water User
Information:
I/O: Output only
Links: Usually not linked

☞ Consumption Shortage

Type: Series Slot
Units: Flow
Description: the total water shortage experienced by the crops
Information: This represents the shortage in terms of the water requested for consumption.
I/O: Output only
Links: Usually not linked

☞ Supplemental Diversion Priority

Type: Table Slot
Units: None
Description: an integer value which represents the Water User's priority in supplemental water rights
Information: A low integer value indicates a high priority.
I/O: Required input
Links: Not linkable

☞ Supplement Return Flow

Type: Series Slot
Units: Flow
Description: the amount of return flow from the supplemental water
Information: Calculated by the same method that calculates return flow from surface diversions.

I/O: Output only
Links: May be linked to the Return Flow slot on any object or the Inflow slot on a Groundwater Storage object.

☞ Supplemental Use Delivery

Type: Series Slot
Units: Flow
Description: amount of supplemental water received by the Water User
Information:
I/O: Output only
Links: Usually not linked

☞ Supplemental Use Request

Type: Series Slot
Units: Flow
Description: the amount of supplemental water requested
Information:
I/O: Optional; if not input, it is set to the lower value of Diversion Shortage and Maximum Supplement Request.
Links: Not linkable

☞ Total Applied Water

Type: Series Slot
Units: Flow
Description: total amount of water available for consumption/use by the water user
I/O:
Links: Not linkable

☞ Total Return Flow

Type: Series Slot
Units: Flow
Description: the sum of the Return Flow and the Supplement Return Flow
Information: This value is split into surface and groundwater components if a method is selected in the Return Flow Split category.
I/O: Output only
Links: May be linked to the Return Flow slot on any object or the Inflow slot on a Groundwater Storage object

For both linking structures, Supplemental Use Request is set by the following equation:

$$\text{Supplemental Use Request} = MIN(\text{Diversion Shortage, Maximum Supplement Request})$$

(Maximum Supplement Request slot values are set by the Max Supplemental Request method.)

If the default linking structure is active, Supplemental Diversion is set by the following equation:

$$\text{Supplemental Diversion} = MIN(\text{Available Supplemental Water, Supplemental Use Request})$$

Supplemental Use Delivery is then set equal to Supplemental Diversion.

However, if the sequential linking structure is active, Supplemental Diversion is a slot on the Agg Diversion Site and therefore, is calculated there. The Agg Diversion Site then sets Supplemental Use Delivery on each Water User based on the Supplemental Diversion and the Supplemental Diversion Priority of each Water User. This is explained in more detail on the Agg Diversion Site object.

The following calculations are then performed regardless of the linking structure selected:

$$\text{Total Applied Water} = \text{Diversion} + \text{Supplemental Use Delivery}$$

The Supplement Return Flow is then calculated by the method selected in the Return Flow category. For example, if the Fraction Return Flow method is selected, Supplement Return Flow is the product of Supplemental Use Delivery and the Fraction Return Flow. If the Variable Efficiency method is selected, the efficiency is recalculated based on the Total Applied Water. Return Flow and Depletion are then recalculated for the new Efficiency value. Supplement Return Flow is calculated by the following equation:

$$\text{Supplement Return Flow} = \text{Supplemental Use Delivery} \times (1 - \text{Efficiency})$$

Consumption and Consumption Shortage are then computed.

$$\text{Consumption} = \text{Total Applied Water} - \text{Return Flow} - \text{Supplement Return Flow}$$

$$\text{Consumption Shortage} = \text{Depletion Requested} - \text{Consumption}$$

Finally, compute the Total Return Flow:

- If the **Return Flow** is linked or the **Supplement Return Flow** is linked, the **Total Return Flow** is zero; both slots are linked so the flows go elsewhere.
- If the **Return Flow** is linked but the **Supplement Return Flow** is not linked, the **Total Return Flow** is equal to the **Supplement Return Flow**.
- If the **Return Flow** is not linked and the **Supplement Return Flow** is linked, the **Total Return Flow** is equal to the **Return Flow**.
- If neither **Return Flow** nor the **Supplement Return Flow** is linked, the **Total Return Flow** is the sum of **Return Flow** and **Supplement Return Flow**.

When the Water User is an element of a sequential Agg Diversion Site, the Outgoing Available water is set to the computed Total Return Flow. This determines how much water is available for downstream elements.

27.1.8.3 Supplement Diversion including Soil Moisture

This method allows supplemental diversions when also modeling Soil Moisture. As a result, you must be using both the **Irrigation Requests with Soil Moisture** method, HERE (Section 27.1.1.5), and the **Proportional Shortage with Soil Moisture** method HERE (Section 27.1.4.4) or **Variable Efficiency with Soil Moisture** method HERE (Section 27.1.4.6).

Note, this method is available:

- on a stand alone Water User
- on an element of a No Structure Agg Diversion Site, or
- on an element of a Sequential Agg Diversion Site.

It is not available on a water user on a Lumped Agg Diversion Site.

See also a diagram of the use of Soil Moisture slots HERE (Section 27.1.1.5).

Slots Specific to this Method

☞ Available Supplemental Water

Type: Series Slot
Units: Flow
Description: represents the amount of water available to supplement surface water diversion
Information:
I/O: This value is required for the method to solve. It is usually propagated across a link.
Links: Usually linked to the Available For Pumping slot on a Groundwater Storage object.

☞ Consumption

Type: Series Slot
Units: Flow
Description: Total amount of water consumed/used by the Water User
Information:
I/O: Output only
Links: Usually not linked

☞ Consumption Shortage

Type: Series Slot
Units: Flow
Description: The total water shortage experienced by the crops
Information: This represents the shortage in terms of the water requested for consumption.
I/O: Output only
Links: Usually not linked

☞ Consumptive Use from Supplemental

Type: Series Slot
Units: Flow
Description: The consumed water from the Supplemental Diversion
Information:
I/O: Output Only
Links: Not Linkable

☞ Irrigated Area

Type: Series Slot
Units: Area
Description: surface area of the land to be irrigated
Information:
I/O: Specified according to the selected method in the Irrigation Acreage and Evapotranspiration Rates category HERE (Section 27.1.2).
Links: Not linkable

☞ Maximum Soil Moisture

Type: Scalar
Units: Length
Description: The depth of water that the soil can hold.
Information:
I/O: Required Input
Links: Not linkable

☞ Minimum Efficiency

Type: Series Slot
Units: Fraction
Description: Minimum amount of consumed flow per unit diverted flow
Information: Expressed as a fraction.
I/O: Required input
Links: Not linkable

Links:

☞ Soil Moisture

Type: Series Slot
Units: Volume
Description: This slot tracks the volume of soil moisture water in the soil at each timestep in the run.
Information: For the initial timestep, this slot must be valid. If this slot is not input, it will be computed as the Maximum Soil Moisture times the Irrigated Area at the initial timestep. That is, the soil moisture starts full.
I/O: Typically Output for run timesteps.

Links: Not linkable

☞ SUPPLEMENTAL DIVERSION

Type: Series Slot
Units: FLOW
Description: The amount of water pumped from groundwater or diverted for supplemental use.
Information:
I/O: Input, Output, or set by a rule.
Links: Can be linked to the Pumped Flow on the Groundwater Storage Object.

☞ SUPPLEMENTAL FLOW TO SOIL MOISTURE

Type: Series Slot
Units: FLOW
Description: The portion of the supplemental flow that goes to the soil moisture.
Information: This is only positive if Supplemental Diversion is set (input or rules) to a value larger than Supplemental Use Request and less than the infiltration limit.
I/O: Output Only
Links: Not Linkable

☞ SUPPLEMENTAL REQUEST FRACTION

Type: Series with Periodic Input
Units: FRACTION
Description: The fraction of the remaining demand that should be met with supplemental water.
Information: If not specified, it is assumed to 1.0.
I/O: Optional input as either a series or periodic relationship.

☞ SUPPLEMENT RETURN FLOW

Type: Series Slot
Units: FLOW
Description: The amount of return flow from the supplemental water
Information:
I/O: Output only
Links: May be linked to the Return Flow slot on any object or the Inflow slot on a Groundwater Storage object.

☞ SUPPLEMENTAL RUNOFF

Type: Series Slots
Units: FLOW
Description: The amount of applied supplemental water that runs off the fields directly due to max infiltration limits.

Information: If the Supplemental Runoff slot **is** linked to a slot on another object it is not included in the total Return Flow. If the Supplemental Runoff **is not** linked, the Supplemental Runoff is included as part of the total Return Flow.
I/O: Output only
Links: Linkable

☞ Supplemental Use Delivery

Type: Series Slot
Units: Flow
Description: amount of supplemental water received by the Water User
Information: For the No Structure case, this is always equal to the Supplemental Diversion.
I/O: Output only
Links: Not linked for the No Structure case.

☞ Supplemental Use Request

Type: Series Slot
Units: Flow
Description: the amount of supplemental water requested
Information:
I/O: Optional; If not input, it is set to the lower value of Diversion Shortage or Maximum Supplement Request.
Links: Not linkable

☞ Total Applied Water

Type: Series Slot
Units: Flow
Description: total amount of water available for consumption/use by the water user
Information: It is computed as the sum of Delivered Flow, Excess Effective Precipitation Flow and Supplemental Use Delivery.
I/O: Output only
Links: Not linkable

☞ Total Return Flow

Type: Series Slot
Units: Flow
Description: the sum of the Return Flow and the Supplement Return Flow
Information: This value is split into surface and groundwater components if a method is selected in the Return Flow Split category.
I/O: Output only
Links: May be linked to the Return Flow slot on any object.

The following equations are executed from the dispatch method (this applies to a stand alone or No Structure water user). This method is not available on a sequential Agg Diversion Site, but the soil moisture methods above can be used.

If **Proportional Shortage with Soil Moisture** is selected, efficiencyTemp is set to the **Minimum Efficiency**.

$$\text{efficiencyTemp} = \text{Minimum Efficiency}$$

If **Variable Efficiency with Soil Moisture** is selected, efficiencyTemp is set to the computed efficiency on the slot.

$$\text{efficiencyTemp} = \mathit{Efficiency}$$

If Maximum Supplement Request is in use and valid:

$$\text{Supplemental Use Request} = MIN\left(\frac{\text{Crop Shortage} \times \text{Supplemental Request Fraction}}{\text{efficiencyTemp}},\ \text{Maximum Supplement Request}\right)$$

Otherwise

$$\text{Supplemental Use Request} = \frac{\text{Crop Shortage} \times \text{Supplemental Request Fraction}}{\text{efficiencyTemp}}$$

Since some water has already been applied, compute the new infiltration limit.

$$\text{newInfitrationLimit} = MAX((\text{Max Infiltration Rate} \times \text{Irrigated Area} - \mathit{flowToIrrigatedArea}),\ 0.0cms)$$

If Supplemental Diversion is **not** specified (Input or set by a rule),

$$\text{Supplemental Diversion} = MIN((\text{Available Supplemental Water, Supplemental Use Request, newInfiltrationLimit}))$$

Otherwise, the specified Supplemental Diversion is used.

Compute any Supplemental Diversion that runs off due to the infiltration limit:

$$\text{Supplemental Runoff} = \text{Supplemental Diversion} - \text{newInfitrationLimit}$$

Compute a temporary variable representing the supplemental flow that infiltrates into the soil:

$$\mathit{supplementalInfiltration} = \text{Supplemental Diversion} - \text{Supplemental Runoff}$$

Next, compute the supplemental use delivery.

$$\text{Supplemental Use Delivery} = \text{Supplemental Diversion}$$

$$\text{Total Applied Water} = \text{Delivered Flow} + \text{Excess Effective Precipitation Flow} + \text{Supplemental Use Delivery}$$

If **Variable Efficiency with Soil Moisture** is selected, recompute the efficiencyTemp as the new

$$\text{efficiencyTemp} = MIN\left(\frac{MIN(\text{Depletion Requested, maxInfiltrationFlow})}{MIN(\text{Total Applied Water, Diversion Request for Crops})}, \text{Maximum Efficiency}\right)$$

When **Proportional Shortage with Soil Moisture** is selected, efficiencyTemp continues to be the Minimum Efficiency.

Compute the *initialSupplementReturnFlow*. This is the amount that would become return flow regardless of soil moisture

$$initial\text{SupplementReturnFlow} = supplementalInfiltration \times (1 - efficiencyTemp)$$

Calculate consumed supplemental flow:

$$consumedSupplementalFlow = MIN(\text{Crop Shortage}, supplementalInfiltration - initial\text{SupplementReturnFlow})$$

Calculate remaining available for soil moisture:

$$availableForSoilMoisture = supplementalInfiltration - initial\text{SupplementReturnFlow} - consumedSupplementalFlow$$

Calculate the new limits on soil moisture refill including the flow to fill the soil moisture and the new infiltration limit:

$$\text{flowToFillSoilMoistureWithSupplement} = \frac{\text{Maximum Soil Moisture} \times \text{Irrigated Area} - \text{Soil Moisture}(t)}{\Delta t}$$

Calculate the supplemental flow that fills the soil moisture:

$$\text{Supplemental Flow to Soil Moisture} = MIN(availableForSoilMoisture, \text{flowToFillSoilMoistureWithSupplement}, \text{newInfiltrationLimit})$$

Determine the new Soil Moisture volume:

$$\text{Soil Moisture[t]} = \text{Soil Moisture[t-1]} + \text{Soil Moisture Gain Loss} + (\text{Soil Moisture Flow} + \text{Supplemental Flow to Soil Moisture}) \times \Delta t$$

Calculate final Supplemental Return Flow.

If the Supplemental Runoff slot is linked to another object, it is not included in the total Return Flow:

$$\text{Supplement Return Flow} = supplementalInfiltration - consumedSupplementalFlow - \text{Supplemental Flow to Soil Moisture}$$

Otherwise, the Supplemental Runoff is not linked, the Supplemental Runoff is included as part of the total Supplemental Return Flow:

$$\text{Supplement Return Flow} = \text{Supplemental Use Delivery} - consumedSupplementalFlow - \text{Supplemental Flow to Soil Moisture}$$

Compute Consumption and Consumption Shortage, and set Consumptive Use from Supplemental.

$$\text{Consumption} = Depletion - MIN(0, \text{Soil Moisture Flow}) + consumedSupplementalFlow$$

$$\text{Consumption Shortage} = MAX(\text{Depletion Requested} - \text{Consumption}, 0)$$

$$\text{Consumptive Use from Supplemental} = consumedSupplementalFlow$$

Finally, compute the Total Return Flow:

- If the **Return Flow** is linked or the **Supplement Return Flow** is linked, the **Total Return Flow** is zero; both slots are linked so the flows go elsewhere.
- If the **Return Flow** is linked but the **Supplement Return Flow** is not linked, the **Total Return Flow** is equal to the **Supplement Return Flow**.
- If the **Return Flow** is not linked and the **Supplement Return Flow** is linked, the **Total Return Flow** is equal to the **Return Flow**.
- If neither **Return Flow** nor the **Supplement Return Flow** is linked, the **Total Return Flow** is the sum of **Return Flow** and **Supplement Return Flow**.

When the Water User is an element of a sequential Agg Diversion Site, the Outgoing Available water is set to the computed Total Return Flow. This determines how much water is available for downstream elements.

27.1.9 Max Supplemental Request

This method category is available to Water Users when the Supplement Diversion method in the Conjunctive Use category is selected. It solves for the Maximum Supplement Request slot used in Conjunctive Use calculations.

27.1.9.1 None

This is the default user method. It performs no calculations and has no slots specifically associated with it.

27.1.9.2 Input Max Request

This method instantiates the Maximum Supplement Request slot for user input.

Slots Specific to this Method

☞ **Maximum Supplement Request**

Type: Series Slot
Units: Flow
Description: user input maximum supplemental request
Information:
I/O: Optional, not used if not input
Links: Not linkable

27.1.9.3 GW Elevation Max Request

This method calculates the Maximum Supplement Request slot value based upon groundwater elevations at the previous timestep.

Values for the Previous Groundwater Elevation slot are set somewhat differently depending on the linking structure. When the No Structure linking structure is selected or the Water User is a stand alone object, Previous Groundwater Elevation values are set by a link to a Groundwater Storage object's Previous Groundwater Elevation slot. When the Sequential Structure is selected, a Previous Groundwater Elevation slot also becomes available on the parent Agg Diversion Site object. When this is the case, the user should link the Agg Diversion Site's Previous Groundwater Elevation slot to the Groundwater Storage object. An automatic link exists between the Agg Diversion Site ant its elements so it is not necessary to link the Previous Groundwater Elevation slot on the Water Users for the sequential linking structure.

Slots Specific to this Method

☞ **Max Request Table**

Type: Table Slot

Units: LENGTH VS. FLOW
Description: user input relating groundwater elevation and maximum supplemental request
Information:
I/O: Required input
Links: Not linkable

☞ PREVIOUS GROUNDWATER ELEVATION

Type: Series Slot
Units: LENGTH
Description: previous groundwater elevation propagated across a link
Information: When the sequential linking structure is active, this slot is automatically linked to the Previous Groundwater elevation slot on the Aggregate object. Only the Aggregate object should be linked to the Groundwater Storage object. When the No Structure method is active or the Water User is a stand alone object, this slot should be linked by the user to the Groundwater Storage object
I/O: Optional; usually set by a link
Links: Linked to the Previous Groundwater Elevation slot on Groundwater Storage object if the linking structure is NOT sequential.

☞ MAXIMUM SUPPLEMENT REQUEST

Type: Series Slot
Units: FLOW
Description: maximum supplemental request based upon Max Request Table
Information:
I/O: Output only
Links: Not linkable

Maximum Supplement Request is calculated by interpolation of the Max Request Table using the Previous Groundwater Elevation.

27.1.10 Multiple Supply Sources

The Multiple Supply Sources category contains a method that adds the Supply From Reservoirs slot. This slot is used by a pre-defined RPL function (called ComputeReservoirDiversions HERE (RPLPredefinedFunctions.pdf, Section 20)) that determines diversions from multiple reservoirs to meet the Diversion Requested on a single Water User object. In this case, the reservoir object also usually serve more than one Water User.

27.1.10.1 None

There are no slots or calculation associated with this method

27.1.10.2 Multiple Supply Reservoirs

This method adds the Supply From Reservoirs slot that is used by a RPL pre-defined function (ComputeReservoirDiversions HERE (RPLPredefinedFunctions.pdf, Section 20)) that computes multiple reservoir diversions. There are no calculations associated with this method.

SLOTS SPECIFIC TO THIS METHOD

☞ **SUPPLY FROM RESERVOIRS**

Type: No Compute Multi Slot
Units: FLOW
Description: The delivery from each reservoir that supplies the water user object
Information: A link is created to the Multi Outflow slot on each Diversion Object that connects to a possible supply reservoir. The values in this slot are determined by a pre-defined RPL function (ComputeReservoirDiversions HERE (RPLPredefinedFunctions.pdf, Section 20)).
I/O: Output only
Links: Linked to the Multi Outflow slot on Diversion Objects

☞ **MAXIMUM DELIVERY RATES**

Type: Table Slot
Units: FLOW
Description: The maximum delivery rate associated with each source specified in the Supply From Reservoirs slot
Information: Each column will be dynamically generated to correspond to a subslot/link to the Supply From Reservoir slot. At beginning of run, a check will make sure there is a column and value on this slot for each subslot on the Supply From Reservoirs slot.

27.1.11 MODFLOW Link Category WU

This category is used to specify whether this Water User is linked to an external MODFLOW model.

27.1.11.1 No Link to MODFLOW WU

No computations or slots are associated with the “No Link to MODFLOW WU” method, this is the default method for the “MODFLOW Link Category WU”.

27.1.11.2 Link to MODFLOW WU

Note: RiverWare’s connection with MODFLOW is currently not functional. This method has been disabled and cannot be selected. An error will be posted at model load if this method was previously selected. Contact CADSWES for help.

The “Link to MODFLOW WU” method allows the Water User object to be linked with MODFLOW. No computations or slots are associated with the “Link to MODFLOW WU” method. A surface water body represented in MODFLOW may receive a surface return flow from a RiverWare Water User object. That is, the value on the Surface Return Flow slot may be transferred through the computational subbasin structure to MODFLOW when either of the Return Flow Split methods are selected and the “Link to MODFLOW WU” method is selected. Click HERE (Section 7.2.1) to view the RiverWare - MODFLOW Connection Functionality Guide. A description of the Water User specific data configuration is presented in that guide HERE (Section 7.2.1.3.4).

27.2 Dispatch Methods

The Water User object can dispatch to solve for Diversion or can dispatch when Diversion is given.

NOTE: When the Supply From Reservoir slot is in use, the Supply From Reservoir slot values are set to zero for all timesteps during begin run initialization. This allows the zero values to propagate up to the Diversion object and then to the Diversion slot on the Reservoir. This is done because every reservoir dispatch method requires the Diversion value to be known. By defaulting these values to zero, the reservoir is able to dispatch without waiting for the ComputeReservoirDiversions, HERE (RPLPredefinedFunctions.pdf, Section 20), function to execute and set the actual diversion values on the Supply From Reservoir slot.

If the Return Flow slot is linked to a reservoir object, those values are also defaulted to zero for all timesteps. This is done for the same reason as described above - so the reservoir can dispatch with a zero return flow before the actual return flow value is calculated.

27.2.1 No Structure

The Water User object will only dispatch using these methods when the No Structure linking method is selected on an Agg Diversion Site or when the Water User is a stand alone object.

27.2.1.1 Solve given Diversion Requested

Note: You can typically let the selected Diversion and Depletion Request method set Diversion Requested at the beginning of the run or beginning of the timestep (or input the values). If you wish to set Diversion Requested using a rule, the rule computed Diversion Requested will overwrite the method computed value. The water user will then dispatch "Solve Stand alone given Diversion Requested" if Incoming Available Water is known.

REQUIRED KNOWNS

☞ **INCOMING AVAILABLE WATER, DIVERSION REQUESTED**

REQUIRED UNKNOWNS

☞ **DIVERSION, DEPLETION**

The algorithm is used to determine **Diversion**, **Depletion**, **Return Flow**, **Diversion Shortage** and **Depletion Shortage** when the No Structure linking method is used, or the object is stand-alone.

Diversion is set equal to the minimum of **Incoming Available Water** and **Diversion Requested**. **Return Flow** and **Depletion** are then calculated based on the user method selected in the Return Flow category. **Depletion Shortage** is calculated as the difference between **Depletion Requested** and **Depletion**. **Diversion Shortage** is calculated as the difference between **Diversion** and **Diversion Requested**.

If selected, the conjunctive use calculations are then performed.

Next, the total return flow (Return Flow plus Supplement Return Flow, when in use) are split and/or routed, using the selected methods.

When the Specify Scheduled Requests method is selected, the **Depletion Schedule Cutback** and **Depletion Schedule Shortage** are computed as described HERE (Section 27.1.1.11).

27.2.1.2 Solve given Diversion

Note: You can typically specify the Diversion and the Water User will dispatch given Diversion using this dispatch method. To specify Diversion with a rule, the water user's Incoming Available Water should not be linked or Diversion Requested should be left unknown. If Incoming Available Water is linked, the water user could dispatch "Solve Stand alone given Diversion Requested" depending on the timing of the setting of the required knowns.

REQUIRED KNOWNS

☞ **DIVERSION**

REQUIRED UNKNOWNS

☞ **DEPLETION**

The algorithm is used to determine **Depletion**, **Return Flow**, **Diversion Shortage**, and **Depletion Shortage** when the No Structure linking method is used, or the object is stand-alone.

Return Flow and **Depletion** are calculated based on the user method selected in the Return Flow category. If **Depletion Requested** is known, **Depletion Shortage** is calculated as the difference between **Depletion Requested** and **Depletion**. **Diversion Shortage** is calculated as the difference between **Diversion** and **Diversion Requested**.

If selected, the conjunctive use calculations are then performed.

Next, the total return flow (Return Flow plus Supplement Return Flow, when in use) are split and/or routed, using the selected methods.

When the **Specify Scheduled Requests** method is selected, the **Depletion Schedule Cutback** and **Depletion Schedule Shortage** are computed as described **HERE (Section 27.1.1.11)**.

27.2.1.3 Solve given Depletion Requested

This method is available when the Irrigation Requests with Soil Moisture method is selected.

REQUIRED KNOWNS:

- ☞ **DEPLETION REQUESTED**
- ☞ **INCOMING AVAILABLE WATER**

REQUIRED UNKNOWNS:

- ☞ **DIVERSION**
- ☞ **DEPLETION**

The algorithm is used to determine Diversion, Depletion, Return Flow, Diversion Shortage and Depletion Shortage when the No Structure linking method is used, or the object is stand-alone.

First, execute the per-dispatch portions of the computation to compute the Diversion Requested, Diversion Request for Crops, and Diversion Request for Soil Moisture. If Soil Moisture at the previous timestep is unknown, exit and wait for it to become known.

Diversion is set equal to the minimum of Incoming Available Water and Diversion Requested. Return Flow and Depletion are then calculated based on the user method selected in the Return Flow category. Depletion Shortage is calculated as the difference between Depletion Requested and Depletion. Diversion Shortage is calculated as the difference between Diversion and Diversion Requested.

If selected, the conjunctive use calculations are then performed.

Next, the total return flow (Return Flow plus Supplement Return Flow, when in use) are split and/or routed, using the selected methods.

27.2.2 Sequential Structure

The Water User will only dispatch using these methods when the Sequential Structure linking method is selected on the aggregate diversion site that contains these water user elements.

27.2.2.1 Solve Sequential given Diversion Requested

REQUIRED KNOWNS

☞ **INCOMING AVAILABLE WATER, DIVERSION REQUESTED**

REQUIRED UNKNOWNS

☞ **OUTGOING AVAILABLE WATER, DIVERSION**

If **Incoming Available Water** is greater than **Diversion Requested**, then **Diversion** is set equal to **Diversion Requested** and a temporary variable “unusedFlow” is the difference between **Incoming Available Water** and **Diversion**. If **Incoming Available Water** is less than **Diversion Requested**, **Diversion** is set equal to **Incoming Available Water** and unusedFlow is zero.

The **Depletion Shortage** is calculated as the difference between **Depletion Requested** and **Depletion**. **Diversion Shortage** is calculated as the difference between **Diversion** and **Diversion Requested**.

The **Return Flow** and **Depletion** are calculated based on the selected user method in the **Return Flow** category.

When the Specify Scheduled Requests method is selected, the **Depletion Schedule Cutback** and **Depletion Schedule Shortage** are computed as described **HERE (Section 27.1.1.11)**.

If selected, the Conjunctive Use calculations are then performed.

Next, the total return flow (Return Flow plus Supplement Return Flow, when in use) are split uisng the selected **Return Flow Split** method and/or routed, using the selected **Return Flow Routing** methods.

Next, the portion of the outgoing water that is from Return Flow is computed, using a temporary variable called “outgoingFromRF”. This computation depends on the selected Return Flow Split method and Return Flow Routing. Basically, if the return flow slot (Surface Return Flow, Return Flow, Total Return Flow or Routed Return Flow) is linked, it is going elsewhere and not available to the next element, so it is not included in the outgoingFromRF. If the flow components are not linked, it is considered outgoingFromRF.

Finally, the **Outgoing Available Water** is computed as

$$\text{Outgoing Available Water} = unusedFlow + outgoingFromRF$$

The **Outgoing Available Water** propagates across a link to the **Incoming Available Water** for the next Water User or the **Total Unused Water** of the Agg Diversion Site.

27.2.2.2 Solve Sequential given Diversion

Required Knowns

☞ **Incoming Available Water**

☞ **Diversion**

Required Unknowns

☞ **Outgoing Available Water**

If **Incoming Available Water** is greater than **Diversion**, then a temporary variable "unusedFlow" is the difference between **Incoming Available Water** and **Diversion**. If **Incoming Available Water** is less than **Diversion**, unusedFlow is zero.

If **Depletion Requested** is valid, the **Depletion Shortage** is calculated as the difference between **Depletion Requested** and **Depletion**. **Diversion Shortage** is calculated as the difference between **Diversion** and **Diversion Requested**.

The **Return Flow** and **Depletion** are calculated based on the selected user method in the Return Flow category. If the return flow is split and neither **Return Flow** nor **Surface Return Flow** are linked, the value of **Surface Return Flow** is added to the existing **Outgoing Available Water**. If the return flow is not split and **Return Flow** is not linked, the value of **Return Flow** is added to the existing **Outgoing Available Water**.

When the Specify Scheduled Requests method is selected, the **Depletion Schedule Cutback** and **Depletion Schedule Shortage** are computed as described **HERE (Section 27.1.1.11)**.

If selected, the Conjunctive Use calculations are then performed.

Next, the total return flow (Return Flow plus Supplement Return Flow, when in use) are split uisng the selected **Return Flow Split** method and/or routed, using the selected **Return Flow Routing** methods.

Next, the portion of the outgoing water that is from Return Flow is computed, using a temporary variable called "outgoingFromRF". This computation depends on the selected Return Flow Split method and Return Flow Routing. Basically, if the return flow slot (Surface Return Flow, Return Flow, Total Return Flow or Routed Return Flow) is linked, it is going elsewhere and not available to the next element, so it is not included in the outgoingFromRF. If the flow components are not linked, it is considered outgoingFromRF.

Finally, the **Outgoing Available Water** is computed as

$$\text{Outgoing Available Water} = unusedFlow + outgoingFromRF$$

The **Outgoing Available Water** propagates across a link to the **Incoming Available Water** for the next Water User or the **Total Unused Water** of the Agg Diversion Site.

27.2.2.3 Solve Sequential given Depletion Requested

This method is available if the water user has the Irrigation Requests with Soil Moisture method selected.

REQUIRED KNOWNS:

- **DEPLETION REQUESTED,**
- **INCOMING AVAILABLE WATER**

REQUIRED UNKNOWNS:

- **DIVERSION,**
- **OUTGOING AVAILABLE WATER**

The algorithm is used to determine Diversion, Depletion, Return Flow, Diversion Shortage and Depletion Shortage when the Sequential linking method is used.

If Irrigation Requests with Soil Moisture is selected, compute the Diversion Requested, Diversion Request for Crops, and Diversion Request for Soil Moisture. If Soil Moisture at the previous timestep is unknown, exit and wait for it to become known.

If **Incoming Available Water** is greater than **Diversion Requested**, then **Diversion** is set equal to **Diversion Requested** and a temporary variable “unusedFlow” is the difference between **Incoming Available Water** and **Diversion**. If **Incoming Available Water** is less than **Diversion Requested**, **Diversion** is set equal to **Incoming Available Water** and unusedFlow is zero.

The **Depletion Shortage** is calculated as the difference between **Depletion Requested** and **Depletion**. **Diversion Shortage** is calculated as the difference between **Diversion** and **Diversion Requested**.

If selected, the **Conjunctive Use** calculations are then performed.

Next, the total return flow (Return Flow plus Supplement Return Flow, when in use) are split uisng the selected **Return Flow Split** method and/or routed, using the selected **Return Flow Routing** methods.

Next, the portion of the outgoing water that is from Return Flow is computed, using a temporary variable called “outgoingFromRF”. This computation depends on the selected **Return Flow Split** method and **Return Flow Routing**. Basically, if the return flow slot (Surface Return Flow, Return Flow, Total Return Flow or Routed Return Flow) is linked, it is going elsewhere and not available to the next element, so it is not included in the outgoingFromRF. If the flow components are not linked, it is considered outgoingFromRF.

Finally, the **Outgoing Available Water** is computed as

$$\text{Outgoing Available Water} = unusedFlow + outgoingFromRF$$

The **Outgoing Available Water** propagates across a link to the **Incoming Available Water** for the next Water User or the **Total Unused Water** of the Agg Diversion Site.

27.2.3 Lumped Structure

The Water User does not dispatch when the Lumped Structure linking method is selected on the Agg Diversion Site. But, when the Agg Diversion Site dispatches, it will set the **Depletion Shortage** on the element water users. For more information, click **HERE (Section 3.2.4)**. Any shortage on the aggregate is shared by each water user element proportional to its **Depletion Requested**. When the Specify Scheduled Requests method is selected, the **Depletion Schedule Cutback** and **Depletion Schedule Shortage** are also set on the water user elements.

28. Appendix A: Reservoir Convergence

28.1 Introduction

This document describes the convergence algorithm used by the getMaxOutflow, getMaxRelease, getMinSpillGivenInflowRelease and the mass balance functions. These functions are used by the Storage, Level Power and Slope Power reservoirs. The purpose of each of these functions is to solve for the variable of interest which is dependent upon a second changing variable. The functions must be simultaneously solved to determine the variable of interest. For example, the getMaxOutflow function returns the maximum allowable outflow based on the pool elevation. Two function are used in this iterative process; the mass balance and maximum outflow functions. The reservoir inflows are known and the function searches for the solution for maximum outflow that satisfies both the maximum outflow tables (max outflow as a function of pool elevation) and the mass balance equation for the reservoir. As guesses for the maximum outflow change, the pool elevation fluctuates as a result of the changing storage as calculated by the mass balance equation. This pool elevation is then used to determine a new maximum outflow value. Thus, an iterative search algorithm is needed to simultaneously solve the functions. Two types of search algorithms are employed to find the solution. The first algorithm iterates by updating the variable of interest based on the independent variable (usually pool elevation) in each iteration. For example, during each iteration in the getMaxOutflow function, outflow is calculated based on the mass balance pool elevation. At the end of each iteration, the outflow value is compared to that of the previous iteration and the algorithm is exited if these values are within convergence. The values are considered converged if

$$\left|\frac{(\text{Old Value} - \text{New Value})}{\text{New Value}}\right| \times 100 \leq \text{Convergence Percentage}$$

Note, if New Value is zero, the Old Value is used as the denominator. If Old and New Value are both zero, it has converged. Convergence Percentage is a general slot on the reservoir which defaults to 0.0001 if not input by the user.

Under certain circumstances, this simplistic iterative routine may fall into a loop that iterates around the solution and never converges. In this event, the bisection routine is invoked using the final values from the first routine as its initial conditions.

The bisection routine is an incremental search method in which the functions are reevaluated at the midpoint of the interval between the values of the previous guess to determine on which side of the midpoint the root lies. Depending on which side of the midpoint the root lies, the midpoint becomes either the upper or lower bound to the search interval. The interval is again divided in half and the functions are reevaluated at the midpoint. This procedure is repeated until the solution is obtained.

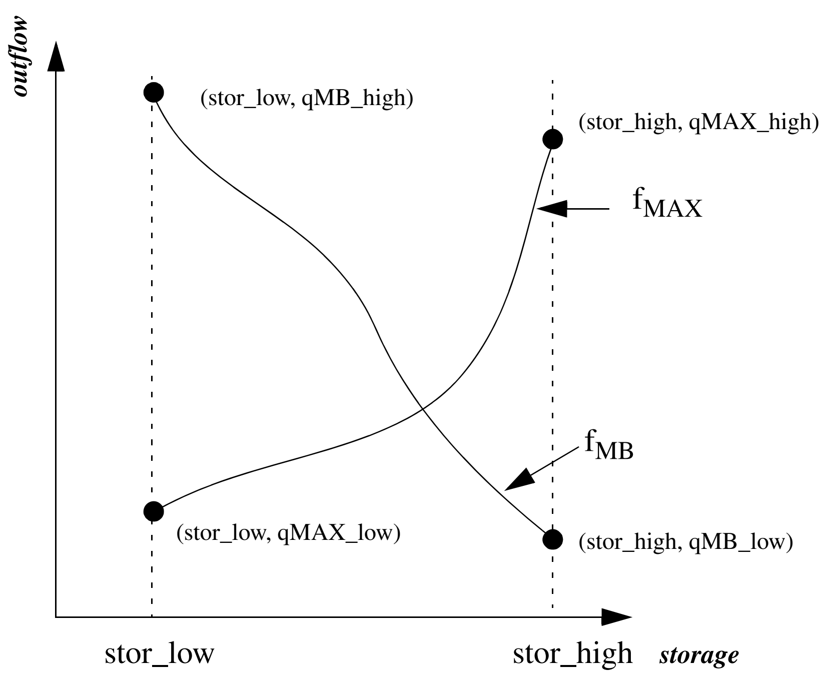
28.2 Example

The following example describes how the bisection method is implemented in the function getMaxOutflowGivenInflow. The mass balance functions and getMinSpillGivenInflowRelease use the identical procedure

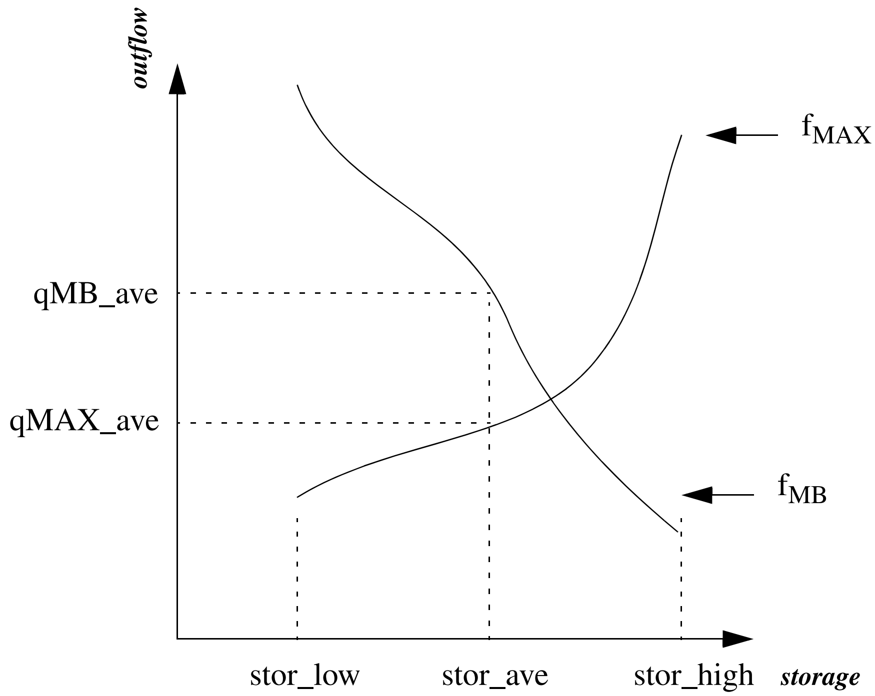
Given the total inflow over the timestep, getMaxOutflowGivenInflow returns the maximum outflow. This function is either called by a rule or executed as a result of a max capacity flag being set on an outflow slot. The maximum outflow is a combination of the maximum release (turbine release if the object is a Level Power Reservoir or a Slope Power Reservoir), the maximum regulated and unregulated spill, and the maximum bypass. In the event that the first iterative routine fails, the bisection routine is started with six final values that were determined during the final two iterations of the first routine. These six values can be combined to form four pairs of Cartesian coordinates where storage is on the x-axis and outflow is on the y-axis. For the purpose of this example, these values are denoted as

stor_high = the largest storage value returned from the mass balance function
stor_low = the smallest storage value returned from the mass balance function
qMB_high = the outflow corresponding to stor_high determined by mass balance
qMB_low = the outflow corresponding to stor_low determined by mass balance
qMAX_high = the outflow corresponding to stor_high determined by the max outflow function
qMAX_low = the outflow corresponding to stor_low determined by the max outflow function

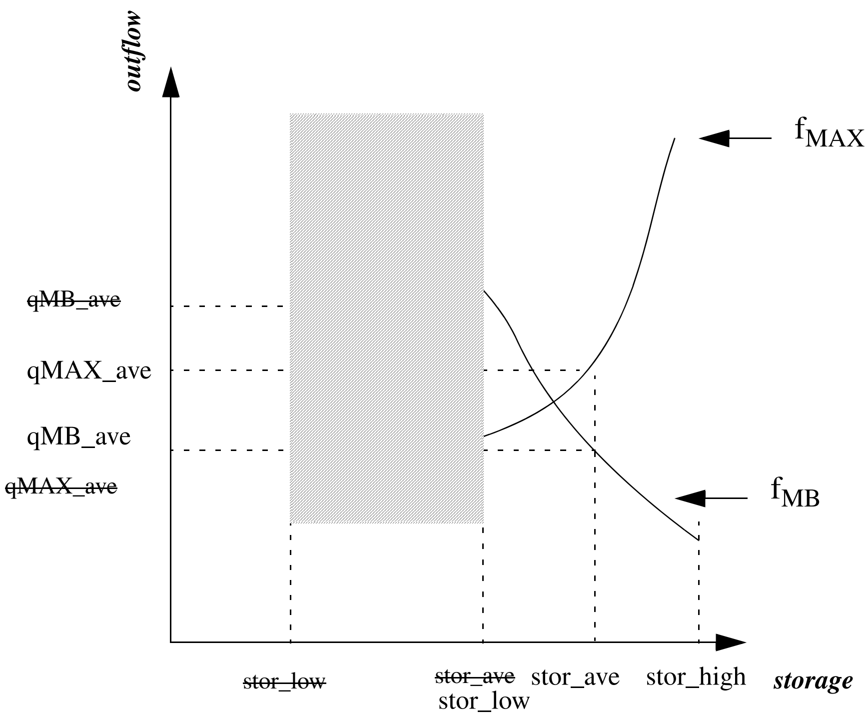
These coordinates are depicted graphically in the figure below. The functions f_{MB} and f_{MAX} denote the mass balance and the max outflow functions, respectively.



The root of the mass balance and max outflow functions is the intersection of the two curves. To begin the search for this intersection, the routine finds the average of stor_high and stor_low, denoted as stor_ave, and then calculates the corresponding outflows, denoted as qMB_ave and qMAX_ave.



Thus, two intervals have been created. The upper interval is between stor_ave and stor_high, the lower interval is between stor_low and stor_ave. Visually it is apparent that the solution lies in the upper interval. The algorithm determines the interval in which the solution lies using an if-statement. Generally, the if-statements compare qMB_ave to qMAX_ave. If qMB_ave > qMAX_ave, the solution lies in the upper interval. Conversely, if qMB_ave < qMAX_ave, the solution lies in the lower interval. In this example, qMB_ave > qMAX_ave, thus, the solution lies in the upper interval and stor_ave becomes the lower bound by setting stor_low equal to stor_ave. The first step of the second iteration is to calculate a new stor_ave to determine the midpoint of the new interval. The figure below illustrates the results of the second iteration.



The search area has decreased because the lower interval was eliminated in the first iteration. The outflows were evaluated using stor_ave resulting in qMB_ave < qMAX_ave. Thus, the solution lies in the lower interval; stor_high is set equal to stor_ave and the third iteration is begun. The first step of the third iteration is again to calculate the midpoint of the new interval. The same process is repeated until the values of qMB_ave and qMAX_ave are within convergence. At this point the algorithm is exited and stor_ave and the minimum value of qMAX_ave and qMB_ave are returned as the solution.

The getMinSpillGivenInflowRelease and the mass balance functions evaluate the variables using different curves, but follow the identical procedure as described above. For example, in getMinSpillGivenInflowRelease f_{max} would be replaced by f_{min} which describes the behavior of the spill curve.

29. Appendix B: Table Interpolation

This document describes the algorithms employed by RiverWare for approximating continuous functions using linear interpolation between a sample of points defining the function. These data are input by the user into data tables (TableSlots), thus we refer to this approach to function approximation as **table interpolation**.

Table interpolation occurs in the following contexts:

- Execution of some user methods during simulation. For example, the “Plant Efficiency Curve” method in the “Power Calculation” category on power reservoirs performs a 3-dimensional table interpolation on the “Plant Power Table”.
- An Optimization run, as part of the process of approximating nonlinear functions as one or more linear functions. For example, linearizing which used the “Stage Flow Tailwater Table” on power reservoirs.
- A Rules or Accounting run in which a RPL block executes the interpolation function. For example, the user could create a table on a data object representing evaporation as a function of temperature. Within a rule one could use 2-dimensional interpolation to find the evaporation corresponding to any temperature in the range of this table.

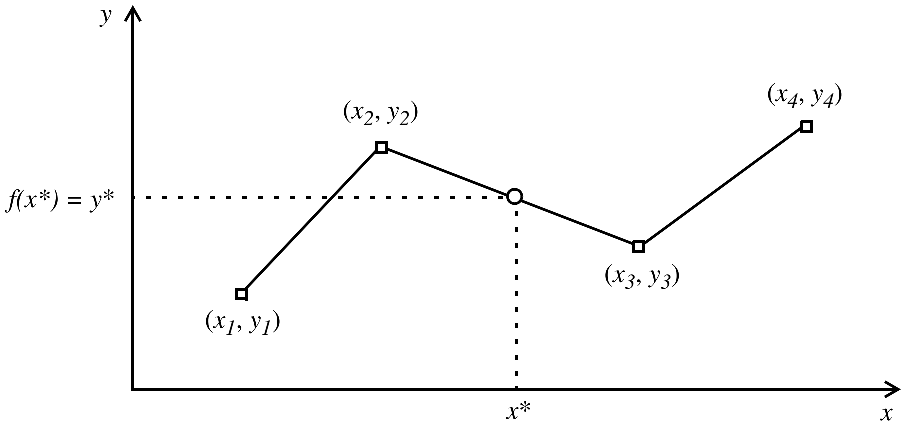
RiverWare supports table interpolation for functions of two and three dimensions. For functions of two variables it is useful to refer to the table columns as x and y, where x is the independent variable and y is the dependent variable: $y = f(x)$. For functions of three variables, we refer to the columns as x, y, and z, where $y = f(x, z)$. We denote a particular approximation using the table by an asterisk: $y^* = f(x^*)$ or $y^* = f(x^*, z^*)$

Two-dimensional data format: For 2-dimensional interpolation RiverWare assumes that the values in the x column of the data table are increasing. The table shown below, the Elevation Volume Table from a reservoir, is an example of the proper way to formulate a table for two dimensional interpolation.

Pool Elevation (ft)	Storage (acre-ft)
440	439,400
441	455,900
442	472,600
443	489,600
445	507,000

Two-dimensional table interpolation: For two-dimensional functions, we apply linear interpolation between data points. The following figure illustrates this approach.

Figure: 4 Two-dimensional linear interpolation



The following types of errors may be reported during two-dimensional table interpolation:

- invalid value (data error): an *x* or *y* value is invalid ($x_i = NaN$ or $y_i = NaN$, for some *i*).
- non-increasing *x* (data error): the *x* values are not increasing ($x_i >= x_{i-1}$, for some *i*).
- out of range (interpolation error): the *x* value being interpolated is out of the range of the table, that is, the domain of the function being approximated ($x^* < x_{min}$ or $x^* > x_{max}$).

Three-dimensional data format: For 3-dimensional interpolation, the z values define blocks: each block has a constant z value and increasing x values, and the blocks are arranged in order of increasing z value. In other words, the three dimensional surface is represented by multiple slices or contours in the x-y plane, each of which may be represented by any arbitrary number of data points, just as with ordinary two-dimensional curves. The table shown below, the Plant Power Table from a power reservoir, is an example of the proper way to formulate a table for three dimensional interpolation.

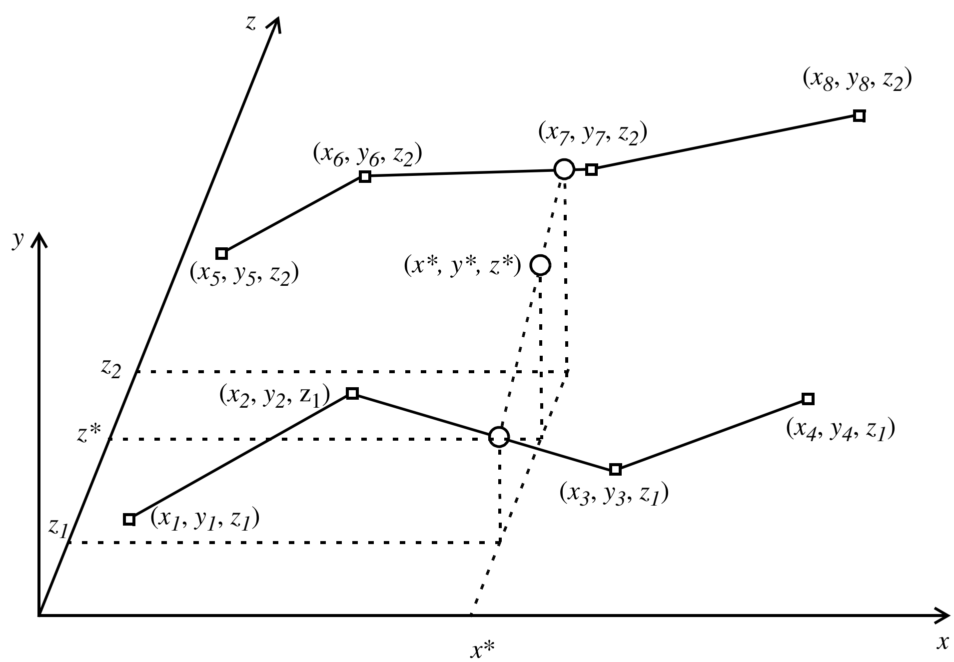
Operating Head (ft)	Turbine Release (cfs)	Power (kW)
100	0	0
100	10	2000
100	20	3000
100	30	4000
200	0	0
200	10	2500
200	20	3500
200	25	3800
200	30	4500
300	0	0

Operating Head (ft)	Turbine Release (cfs)	Power (kW)
300	10	3000
300	25	5000

Three-dimensional table interpolation: For three-dimensional functions, the algorithm for interpolation has two basic cases. If the z value being interpolated is equal to the z value for one of the blocks in the table, then we just perform a two-dimensional interpolation along the curve represented by that block.

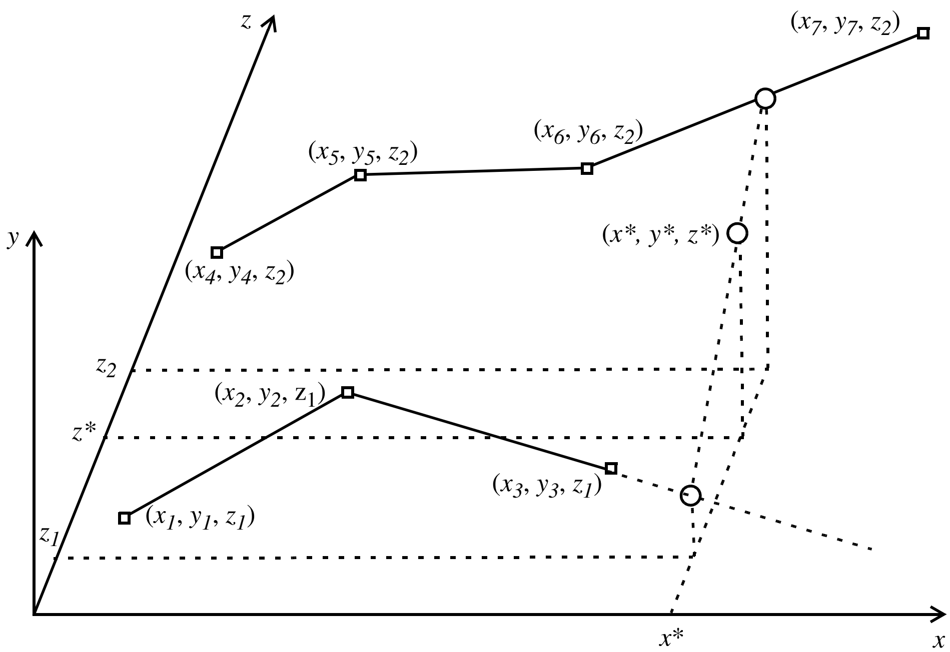
When the z value is not exactly equal to any of the z values found in the table, RiverWare first identifies the constant z-blocks whose values bound the z value being interpolated and performs a two-dimensional interpolation along these curves. This yields two points, one on each bounding constant z-curve, and the final answer is computed by a linear interpolation between these two points. The following figure illustrates this case.

Figure: 5 Three-dimensional linear interpolation



There is one special case in which the interpolation behavior is slightly different: when the x value being interpolated is within the domain of one of the bracketing constant z curves but not the other. In this case, we interpolate between the encompassing curve and the extrapolation of the other (shorter) curve. We extrapolate this curve with either the slope of its last segment or the slope of the corresponding segment of the encompassing curve, as appropriate for the particular table. To avoid over-ambitious extrapolation, RiverWare requires that the answer lie in the region bounded by the constant-z curves (i.e., their convex hull). The following figure illustrates this case, where the short curve is extrapolated with the slope of its last segment.

Figure: 6 Three-dimensional linear interpolation



The following types of errors may be reported during three-dimensional table interpolation:

- invalid value (data error): an *x*, *y*, or *z* value is invalid ($x_i = NaN$, $y_i = NaN$, or $z_i = NaN$ for some *i*).
- non-increasing z (data error): the z values are not increasing for one block to another ($z_i >= z_{i-1}$, for some *i*).
- non-increasing *x* (data error): the *x* values are not increasing ($x_i >= x_{i-1}$, for some *i*).
- z value out of range (interpolation error): the *z* value being interpolated is out of the range of the table ($z^* < z_{min}$ or $z^* > z_{max}$).
- x value out of range (interpolation error): the *x* value being interpolated is out of the domain of both of the two bounding constant z-curves.