RiverWare

Technical Documentation Version 7.3

USACE-SWD Methods

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

USACE-SWD Methods
Table of Contents

USACE-SWD Methods - Overview

This document describes functionality developed to replicate algorithms used by the U.S. Army Corp of Engineers Southwest Division (USACE-SWD). These algorithms, consisting of object methods, RPL predefined functions, and other utilities were developed as a replacement tool for the USACE-SWD's fortran SUPER program. Although the algorithms were implemented for the USACE-SWD, they are general in nature and available to all RiverWare users.

The purpose of this document is to consolidate all of the documentation specifically developed and delivered to the USACE-SWD to describe the algorithms. This document contains documentation and implementation help specific to the USACE-SWD model setup and approach. But, it contains links to the rest of the RiverWare help to describe the general methods and functions. The document first presents an overview of the approach used. Then it describes the steps necessary to set up a basic simulation model and run it as a calibration. Next, there is a description of the process used to define operating policy using a RPL ruleset. Each USACE-SWD policy is then described including the necessary method selections, how to define the policy in RPL, and detailed description of the algorithm. Finally, sections are provided to describe how to set up a yield study, use the Data Management Interface, use output and analysis tools, and how to improve performance of the modeling runs.

This document is intended to guide a user from a blank workspace through adding objects, implementing policy, and finally implementing a yield study.

1. Conceptual Overview

The integration of USACE-SWD algorithms in RiverWare brings together RiverWare's object-oriented modeling features, the power and flexibility of the priority rulebased simulation, and the Computational Subbasin, that allows the execution of multi-object computations from a rule function. The algorithms are thus implemented in modular, object-specific contexts for ease of maintenance and extension, as well as flexibility of use through user selectable methods. Following is a brief overview of the modeling steps and components.

1.1 Modeling Steps

The following diagram, Figure 1, shows the typical modeling steps to implement the functionality described in this document. An overview of each step is described in the following sections and the remainder of this document describes these items in detail.

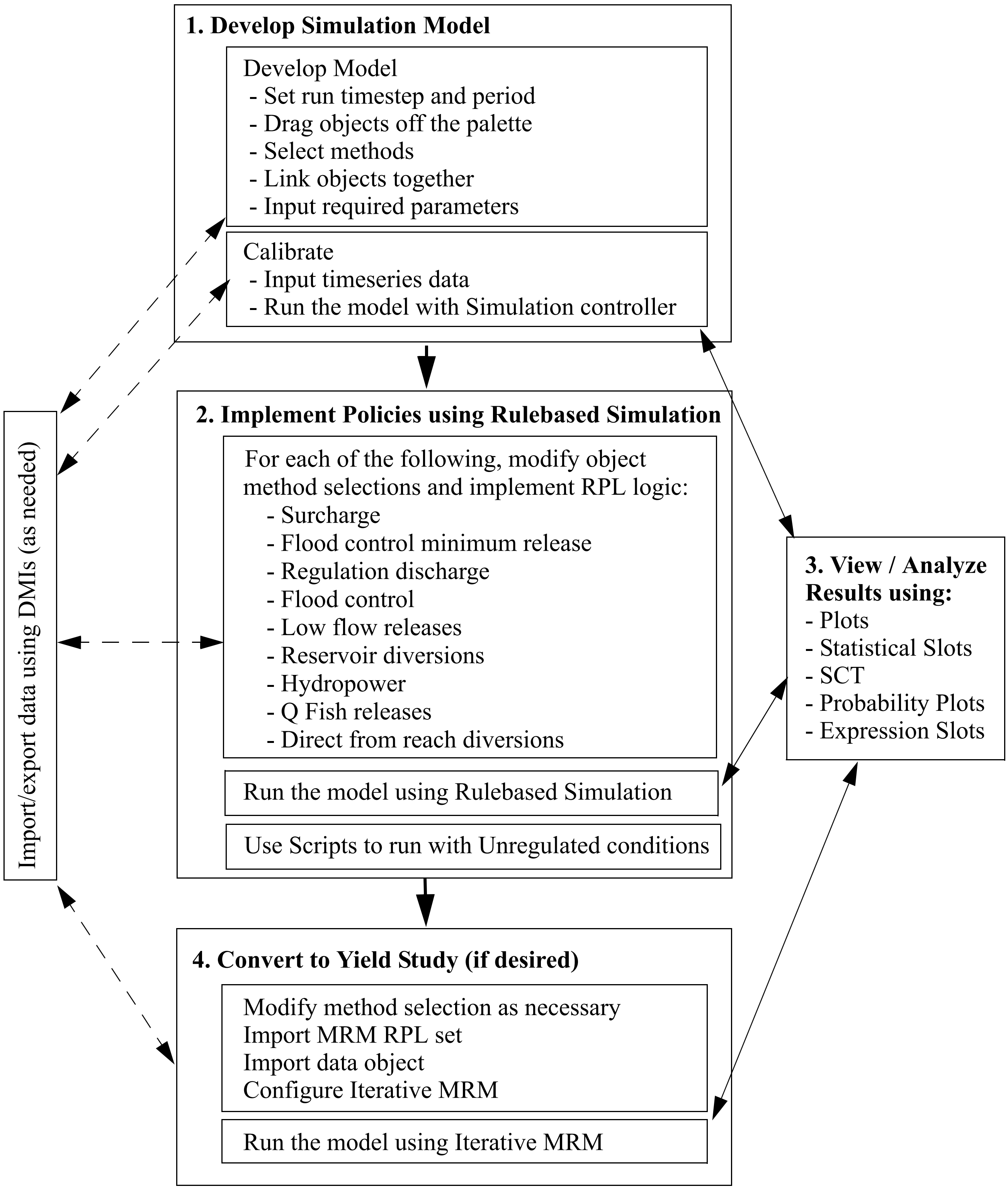


Figure 1. Diagram of Modeling Steps

1.1.1 Develop Simulation Model

The first step is to develop a simulation model of the system by pulling objects off the palette, selecting the appropriate methods, linking objects together and specifying required data. Next, timeseries data is specified. The data management interface (DMI) can be used to import data as desired. With enough data, the model will run. At any point, the output utilities can be used to analyze and view results. The model should be calibrated to make sure that the inputs are producing the desired outputs. For example, you may input inflows to the system and reservoir outflows. You would then make sure that reservoir pool elevation or storage are correct and that downstream flows are routed correctly.

Developing the simulation model is described **HERE (Section 2)**.

1.1.2 Implement policies using Rulebased Simulation

The model can then be converted to a predictive mode by removing some of the inputs (E.g. Storage, Pool Elevation, and Outflows) to under-determine it. Then, rules can be implemented to compute the reservoir outflows based on the state of the system and the operating policies. Described in this document are the following policies: surcharge, regulation discharge, flood control, low flow releases, reservoir diversions, hydropower releases, and fish releases. Each policy may have additional methods to be selected and data to be input. Then, rules can be written in the RiverWare Policy Language (RPL). Many of the rules use specific predefined functions that execute methods on objects.

Implementing policies using rulebased simulation is described **HERE (Section 3)**.

1.1.3 Run using Unregulated Conditions

Sometimes, it is desired to run the model and compute the flows that would have occurred without one or more reservoirs in place. To do so, use scripts to change the model to use unregulated conditions. Use a script to make a run and generate snapshots of desired slots.

Information on defining and running an unregulated system can be found **HERE (Section 4)**.

1.1.4 View and Analyze Results

Given the model and the ruleset, a run is made. There are numerous tools and utilities that can be used to view and analyze results including:

- Plotting: present a graphical view of the data.
- Statistical slots: create frequency duration or exceedence curves. Then, plot them on a probability scale.
- Expression slots: create user defined expressions involving slot or other values. For example, this can be used for aggregation of data.
- Data Management Interface: DMI’s can be used to export data to an external sink.
- System Control Table: SCT’s can be used to view and edit data in a spreadsheet like format.

Viewing and Analyzing results is described **HERE (Section 5)**.

1.1.5 Convert to Yield Study if desired

A yield study is used to determine the largest constant diversion that can be made from a single reservoir such that the reservoir pool elevation will not drop down below the bottom of conservation pool at any time during the run period. Other operating policies like Surcharge, Regulation Discharge, Flood Control, Low Flow Release, and Hydropower are included in this analysis.

The Yield study uses the iterative Multiple Run Management configuration. In this configuration, the model is run multiple times with a different trial value for the diversion from a reservoir. After each run, logic is used to determine if the yield has been met and if not, a new trial value is calculated and a new run is made. Multiple reservoir can be included; each reservoir's yield is computed separately, starting at the top of the system. Once the first reservoir's yield is found, the next downstream reservoir's yield is found. Thus, the model is run multiple times to find the yield from one reservoir, then that reservoir's yield is "locked in" and the analysis is repeated on the next reservoir.

An existing model is converted to a yield study by importing data and an MRM RPL set. Then, the user sets up an MRM configuration and specifies the search algorithm (Bisection or a heuristic approach). The user can also specify the convergence criteria including the initial trial values and the maximum iterations. Finally, the run is made. The SCT can then be used to view the results of the multiple runs. Included in the results (for each reservoir) is the yield, the critical draw down date, the critical draw down period and the critical period duration.

The conversion to a yield study is a separate modeling step that may or may not be required for a given basin or portion of a basin. Because it is a self contained step, all of the documentation is provided **HERE (Section 7)** and is not discussed further in this overview section.

1.2 Model Components

Implementation of the USACE-SWD functionality makes use of many of RiverWare's modeling features. Following is a description of the various pieces:

1. Methods on objects

On reservoirs, there are user selectable methods to specify how to calculate the incremental flows, forecast hydrology, surcharge release (mandatory releases), flood control, low flow releases, reservoir diversions, and hydropower method.

Control Point objects are used in the basin network to represent channel control points that influence flood control and other releases. User selectable methods on Control Point objects support the calculation of forecasted inflows (uncontrolled area flows), regulation discharge, and empty space hydrographs. The user selectable Key Control Point Balancing method allows for the computation of a balance level for associated reservoirs.

The reach, water user, and diversion object also have user selectable methods that are used as part of the USACE-SWD functionality.

2. Computational Subbasin

RiverWare's computational subbasin has object attributes including user selectable method categories, methods, and data slots to support calculations that involve multiple simulation objects simultaneously. The flood control category includes two methods that govern the nature of the flood control solution. The Operating Level Balancing method is used to model flood control based on the USACE-SWD algorithms. The Phase Balancing method is based on the USACE-KC flood control operations. This document deals with the Operating Level Balancing calculations. The Phase Balancing method will not be discussed in this document. In addition to the flood control category, several related categories and methods are available to adjust/modify some of the details of the flood control algorithm.

Additionally, there are user selectable methods to initialize objects within the subbasin, compute incremental flows, computer reservoir diversions, compute low flow releases, and compute alternative routing coefficients within the subbasin.

3. Surcharge Release and Regulation Discharge Flags

The surcharge release and regulation discharge flags are used to trigger the surcharge release and regulation discharge calculations on reservoirs and control points, respectively. Rules are used to set the surcharge release flag on the outflow slot of each reservoir. When this flag is set, it triggers the reservoir to dispatch and compute a surcharge release forecast. The regulation discharge flag is set on all control points simultaneously by a single rule. This flag triggers each control point to solve for its regulation discharge and empty space hydrographs according to the user methods selected on each control point object.

4. Rule functions to implement the policy

Rules execute predefined rule functions that implement the policy. For example, a predefined rule function called FloodControl invokes the computational subbasin method and returns the solution by way of a list of object.slot and value pairs to be set by the rule. The FloodControl function executes the currently selected method in the Flood Control category on the computational subbasin object. The method is executed on the subbasin named in an argument passed to the FloodControl rule function.

Additionally, the following predefined functions execute the policy listed:

- MeetLowFlowRequirement: computes reservoir release to meet a demand at a control point
- ComputeReservoirDiversions: computes the diversion from each reservoir to meet water user demands.
- HydropowerRelease: computes the additional release required to meet the specified load without causing additional downstream flooding.

1.3 Ruleset and Policy

The ruleset implements the policy that is used to make reservoir releases and diversions from the system. The following policies apply to one or more USACE-SWD applications. Presented is a conceptual overview. Each piece is described in detail in the document.

1.3.1 Surcharge Release Rules

The surcharge release calculations are executed by the reservoir object when the surcharge release flag is set on the outflow slot by a rule. Each rule sets the surcharge release flag on the outflow slot of a single reservoir object. The rules must start at the upstream end of the basin. When the surcharge release flag is set, the reservoir will dispatch during post-rule simulation if the Inflow to the reservoir is also known. During dispatching, the reservoir will compute the surcharge release forecast and set the Outflow slot for the current timestep and all other timesteps in the forecast period. These outflow values propagate via links to downstream objects and are routed using the selected routing method on any Reach objects that are encountered. When the flow values reach a downstream reservoir, that reservoir will have Inflow slot values for the forecast period. A subsequent rule will set the surcharge release flag on that reservoir resulting in dispatching. In this way, each subsequent downstream reservoir calculates the surcharge releases for the forecast period considering the routed surcharge releases from upstream reservoirs.

1.3.2 Flood Control Minimum Release

Using rule logic, Flood Control Minimum Release values can be set after the surcharge operation. It is up to the user to define the rule logic for these releases. These releases are considered minimums because they are made regardless of what the flood control logic computes. Once set, the releases are allowed to propagate downstream and occupy channel space.

1.3.3 Regulation Discharge Rule

After the above rules have executed and the releases have been routed downstream, a single rule sets the regulation discharge flag on the Reg Discharge Calculation slot for all Control Point objects. This triggers each control point to dispatch and execute the selected regulation discharge methods. Because the uncontrolled area inflow forecasts have already been computed and the surcharge releases have been routed downstream, each control point contains the total discharge at its location for each timestep in the forecast period. This information is used to appropriately compute the regulation discharge and empty space hydrographs.

1.3.4 Flood Control Release Rule

After the regulation discharge rule has executed, the rule to calculate the flood control releases can be executed. This rule calls the pre-defined FloodControl function which executes the selected Flood Control method on the Computational Subbasin. The FloodControl function returns a list of flood control releases that should be made for each reservoir at the current timestep. Also included is the outflow from each reservoir at the current timestep. The outflow is computed as the flood control release plus the surcharge release plus the flood control minimum release. The flood control rule sets the Outflow slot and Flood Control Release slot on each reservoir, at the current timestep, given the values returned by the FloodControl function. When these slots are set, it triggers each reservoir to re-dispatch using the new Outflow value. The reservoir objects re-solve, compute new storage and pool elevation values (as well as execute any user selectable methods), and the new Outflow values are routed downstream.

1.3.5 Low Flow Releases

Low flow demands are specified at control points and each control point has a list of reservoirs that are used to meet a given demand. Also, each reservoir has a specified maximum delivery rate for meeting a low flow requirement.

Once flood control is complete, then the Low Flow rule executes and calls the MeetLowFlowRequirements predefined function. This function determines the low flow release for each reservoir serving a low flow demand (i.e. a control point) as follows: If there is a low flow shortage, the serving reservoirs are sorted by level in descending order. Each reservoir (beginning with the most full reservoir) makes a release until the requirement is met, the maximum low flow release on the reservoir is met, or the reservoir reaches the bottom of the conservation pool (whichever value is lowest).

When the reservoir Outflows are set, the system re-solves to route the flows downstream. The low flow shortages will be recomputed to reflect the low flow releases. The reservoir Operating Levels will be recomputed to reflect the new releases. Then the next rule will execute to determine the low flow releases for the next control point.

1.3.6 Reservoir Diversion

Once low flow releases have been made for all reservoirs, direct from reservoir diversions are calculated by calling the predefined function ComputeReservoirDiversions. Reservoir demands can be a water user or another reservoir. Each demand can draw from multiple reservoirs and each reservoir can act as a source for multiple water users or demand points.

The diversions from the reservoir are modeled using diversion objects while the actual demands are modelled with water user objects that compute their Diversion Request (water supply requirement). Even if the demand point is another reservoir, the demand is modelled with a Water User object (configured to have a 100% return flow which could then be sent to the demand reservoir).

When called, the ComputeReservoirDiversions function computes the diversion to each water user by visiting each reservoir (in order of highest level first) and trying to meet the diversion request but limited by maximum delivery amounts, not drawing below the conservation pool. Also, if the demand is a based on reservoir level, no diversions are made if the demand reservoir and has a higher level than the supply reservoir, or the demand reservoir is in the flood pool.

The rule sets the Supply From Reservoirs and the Incoming Available Water slots causing the water user object to solve. It then propagates values to the diversion object and the reservoir which also dispatch and solve.

1.3.7 Hydropower

After surcharge releases, flood control, low flow releases, and reservoir diversions have been made, a reservoir may release additional water from its power pool to meet a specified power commitment, i.e., the *Load* specified for that reservoir. The hydropower rule executes the HydropowerRelease predefined function. This function tries to meet the specified or computed load but is limited to prevent:

- the pool elevation from dropping below the minimum power pool elevation
- releasing more than generating capacity
- the pool elevation from dropping more than the maximum allowable power pool draw down
- additional downstream flooding.

The rule sets the additional hydropower release and increases the reservoir Outflow. This causes the reservoir to re-dispatch and propagates the water downstream.

1.3.8 Fish Release

On a power reservoir, during the hours of the day when power is not generated, water may be released to meet minimum fish flows. These are called Fish Releases or QFish releases. The release of water to meet a downstream demand is determined based on the available fish water in the reservoir, the other off peak releases that are occurring and the monthly minimum fish flow demand. The fish storage in the reservoir is accounted for each timestep. Fish water is either released or is lost due to evaporation. It is replenished through three methods: all inflow is shared, the accumulated local inflow is shared, or no inflow is shared. Also, anytime the pool exceeds the top of conservation pool, the fish storage is replenished.

1.4 Rule-Simulation Interaction for each Timestep

To summarize, the following are the steps that are taken for each timestep in the model:

1. At the beginning of the timestep, cumulative local inflows are disaggregated spatially and an inflow forecast is computed for each reservoir and control point object.

2. The surcharge release rules execute in upstream to downstream order and set the surcharge release flag on the Outflow slot of each reservoir. After each rule executes, the affected reservoir dispatches and computes its surcharge release forecast. These releases are routed downstream to the Inflow slot of the next reservoir before the next rule executes.

3. The flood control minimum release rules optionally set the reservoir outflow and flood control minimum release based on the rule logic. After each rule executes, the affected reservoir dispatches. These releases are routed downstream before the next rule executes.

4. The regulation discharge rule executes and sets the regulation discharge flag on the Reg Discharge Calculation slot on each control point. The control points dispatch and compute regulation discharge and empty space hydrographs for the forecast period.

5. The flood control rule executes and invokes the FloodControl pre-defined function. This function computes the flood control releases for the subbasin and returns the flood control release and outflow values, for the current timestep, for each reservoir. The flood control rule sets the flood control release and outflow values on each reservoir. The reservoir objects re-dispatch and solve for pool elevation, storage and any other reservoir methods that have been selected. The outflow values are routed downstream.

6. For each Low Flow Release control point

- The low flow release rule execute and invoke the MeetLowFlowRequirements pre-defined function. This function computes the release from each reservoir in the subbasin to meet the low flow demand at the control point. The rule sets the Low Flow Release and Outflow slots causing the objects to dispatch and propagate the effect.
- The low flow diversion rules execute to route any low flow releases around lagging reaches by setting Diversion Requests equal to the low flow release. This causes the reaches and water users to re-dispatch. Thus, low flow releases arrive at their control point destinations immediately.

7. The reservoir diversion rules execute and invoke the ComputeReservoirDiversions predefined function. This function computes the diversion that must be made from each reservoir in the subbasin to meet the demands in the subbasin. Note, a single reservoir can serve multiple demands and a demand can be met from multiple reservoirs. After setting the diversions, the reservoir, diversion objects and water users re-dispatch.

8. The hydropower rule executes the HydropowerRelease predefined function. This function executes the hydropower release function on the flood control subbasin to determine the additional hydropower release that is necessary to meet the load without causing additional downstream flooding. The rule sets the Additional Hydropower Release slot and resets the reservoir Outflow slot. This causes the system to re-dispatch and solve the entire system.

9. The Fish Release rules (where defined) execute and determine the additional outflow to meet the Fish Release demand. The rule sets the reservoir Outflow and reservoir Off Peak Spill slots. This again causes the system to re-dispatch and solve. The rule also performs accounting actions to calculate and set the inflows and outflows to the fish account on the reservoir.

10. If no other rules exist, the timestep is complete. RiverWare moves to the next controller timestep and the process is repeated.

2. Developing the Simulation Model

Before implementing the various USACE-SWD policies, it is necessary to build and calibrate a model of the physical river basin network. This involves selecting objects and linking them together to accurately represent the physical features of the river basin. In addition, user methods are selected to model various physical processes (i.e. spill, evaporation, seepage, routing, etc.) depending on the level of modeling detail required.

Initially, the model should be run in Simulation mode. In Simulation, RiverWare solves an exactly determined system given a set of user inputs. No rules or policy constraints are involved in the solution. Each object performs a mass balance calculation and solves any user selected methods. The purpose of the simulation model is to ensure that the physical features of the basin are correctly modeled. In this stage of model development, the modeler checks input data for accuracy, checks for proper mass balance and routing, and calibrates the model. Only after the physical model has been calibrated, should the modeler begin implementing policy through the use of rules.

2.1 Model Building Steps

Following is a very general description of the steps involved in building a simulation model. These steps should be performed in order with the exception of 5, and 6 which are interchangeable. Each step is discussed in greater detail below.

1. Set the model timestep size and the start/end dates in the Run Control dialog.
2. Set the **Number of Post-Run Dispatch Timesteps.**
3. Create a Unit Scheme
4. Add objects to the workspace from the palette.
5. Select the appropriate methods on each object.
6. Link the objects together.
7. Input the required data.
8. Run the model and evaluate the output.

Additional information on model building can be found **HERE (ModelBuilding.pdf, Section 1)**

2.1.1 Set Timestep and Start Date

Before adding objects to the workspace, the user should select the model timestep and specify the start and end dates of the run. This is done through the Run Control dialog.

2.1.2 Set Number of Post-Run Dispatch Timesteps

As described **HERE (Section 2.10.4)**, set the number of timestep after the run for which dispatching should be allowed. Typically you can set this equal to the forecast period.

2.1.3 Define Display Units through a Unit Scheme

The first step when setting up a model is specifying the display units for objects taken from the palette. The internal units in RiverWare are SI units but you can define the display units, scale, precision and format by setting up a “Unit Scheme”. In the Unit Scheme you configure these four attributes for each unit type. Then you can define exceptions to the general rule. That is, for a specific slot you may want to show the values in some other display unit. More information on features of Unit Schemes may be found in the reference located **HERE (Units.pdf, Section 2)**.

2.1.4 Add objects to the workspace

Build the model network with appropriate selection of objects from the RiverWare object palette. Include control point objects at the desired locations.

2.1.5 Select methods on objects

On each object, select methods to model various physical processes (i.e. spill, evaporation, seepage, routing, etc.) depending on the level of modeling detail required. For the basic simulation model, these methods are described starting in section 2.2. Remember, these methods are strictly for the calibration model, additional methods will be selected later to implement the flood control, surcharge, and conservation pool operations.

2.1.6 Link Slots

Slots on objects should be linked to form the network representing the basin. For example, Reservoir Outflow should be linked to the downstream control point or reach’s Inflow slot.

2.1.7 Input Data

Supply the required data to the objects. There are two types of data, table or parameter data and mass balance data.

- Table/Scalar data: Specify required table/scalar data to the slots on the objects.

- Mass Balance and series data: The user should specify unregulated local inflows to control point and reservoir objects as described **HERE (Section 2.8)**. In addition, specify enough mass balance data, typically on the reservoir objects, to run the model. This could include inflows and reservoir storage, pool elevations, or outflows as described **HERE (Section 2.2.5.1)**.

2.1.8 Run and View

Once all objects have been configured (method selection and input data) and have been linked together, the model is ready for simulation. When the model is run, it should compute the inflow forecasts, mass balance the reservoir objects, and properly route releases downstream. The purpose of these simulations is to ensure that the modeling of the physical system has been done correctly. During this stage the model should be calibrated if necessary. Any error messages or data problems need to be resolved before moving to policy modeling. Since the inflows, uncontrolled area flows, and storage or pool elevation values have been input, each reservoir should mass balance correctly to solve for outflow. These outflows should also be routed downstream. The modeler can also input reservoir outflows and check the resulting storages and pool elevations. The resulting values computed by RiverWare should be checked against known values for accuracy. Click **HERE (Section 4)** for more information on viewing and analyzing results.

2.2 Reservoir Objects

When selecting a reservoir object to represent a reservoir/dam that has power production capabilities, it is best to select a power reservoir object even if the model will not be used for modeling power initially. It is easy to enable and disable power modeling on a power reservoir object but it is not as easy to change from a storage reservoir to a power reservoir after the model has already been completed.

2.2.1 Required inputs

Input the relationship between pool elevation and storage in the Elevation Volume Table slot. Also, for every reservoir, an initial Pool Elevation or Storage is required.

On storage reservoirs, input the pool elevation versus release information in the Max Release slot (For level power reservoirs, similar data will be entered based on the selected power method as described in the next section).

2.2.2 Power Methods

For level power reservoirs, the user must select a method in the Power Calculation category regardless of whether or not the object will be used for power calculation. If the object will not initially be used to compute power, select the **No Power Turbine Flow** method. When this method is selected the Max Flow Through Turbines slot will be added and should be populated with the relationship between pool elevation and the maximum outflow through the turbines. However, no power calculations will take place. If the user does wish to model power, he/she should select the appropriate power calculation method. At this stage any of the methods can be used, but if releases are to be made using the HydropowerRelease function (**HERE (Section 3.9)**), then the **Peak Power Equation with Off Peak Spill** method should be selected. Details are provided **HERE (Objects.pdf, Section 17.1.1.10)** on the required data.

2.2.3 Tailwater Methods

When a power method is selected, a tailwater method must also be selected. The available methods are described **HERE (Objects.pdf, Section 17.1.7)**. When setting up the tailwater method, there are some important aspects to remember. The Tailwater Base Value is a series slot that is used to link to a downstream reservoir's Pool Elevation. If the Tailwater Base Value is not linked to a downstream reservoir, it is best to leave NaNs in the Tailwater Base Value and have the Tailwater Table include all of the data necessary to calculate the correct tailwater.

This section describes how SUPER tailwater data can be input into the Tailwater Table to mimic SUPER's functionality. In SUPER, there is a variable called the Block Loading Tailwater Elevation (BLTE). This is defined as the tailwater that results when releasing Turbine Capacity with the reservoir at the top of the power pool with no other releases. In this description, this is called the "Turbine Capacity at Top of Power Pool."

The Tailwater Table can be input such that the look up flow will result in the correct tailwater elevation. The table will be as follows: If the flow is less than or equal to the Turbine Capacity at Top of Power Pool, then the tailwater is a constant value equal to the BLTE. If the flow is greater than the turbine capacity at top of power pool, then the value is from the right portion of the curve. See Figure 2. Based on sample input data, only the right portion of the curve is currently input into SUPER. This method will allow the USACE-SWD to first input the table to mimic the SUPER methodology but in the future, the full table can be input.

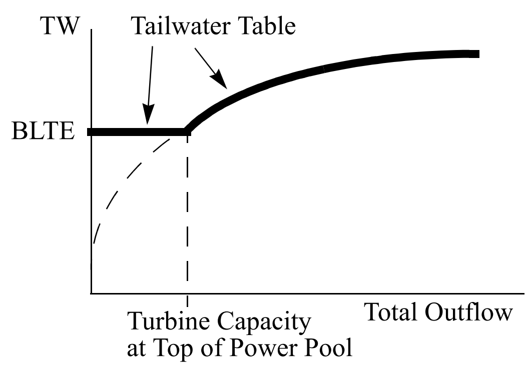


Figure 2. Diagram of Tailwater Table

2.2.4 Spill methods and required data

In RiverWare, a spill is a portion of the outflow that does not go through the turbines (on a power reservoir) or through the main release works (on a storage reservoir). Spills can be controlled (Regulated Spill and Bypass) or uncontrolled (Unregulated Spill).

2.2.4.1 Spill Calculation Category

The spill calculation category is used to model various spillway structures and configurations. Select a spill method only if the model will differentiate between spillway outflow and outflow through the release gates. If a single curve is used to represent total reservoir outflow as a function of pool elevation, then no spill method is necessary. On a storage reservoir the flow vs. pool elevation data is held in the Max Release slot while on power reservoirs it is held in the power method specific slot. If uncontrolled spillways or additional controlled release structures need to be modeled, select the appropriate spill method. If a spill method is selected, enter flow vs. pool elevation data for the various release and spill structures modeled. Click **HERE (Objects.pdf, Section 24.1.3)** for more information on the spill methods.

If a spill method is selected, input the pool elevation vs. spill information on the appropriate slots. For USACE-SWD applications, the “Drift” slots are not used and can remain empty.

2.2.4.2 Spillway and Release Modeling

Previously, the USACE-SWD used a single relationship to define the maximum outflow as a function of pool elevation. This relationship is called the free flow rating curve. The free flow rating curve includes outflow through the turbines, regulated spillways, and unregulated spillways. Normally, the proper spill method would be selected to model each of these outflow structures separately. The data in the free flow rating curve would then need to be disaggregated for each outflow structure modeled. Originally, the USACE-SWD decided not to model each outflow structure separately and wanted to leave the free flow rating curve intact (as opposed to disaggregating the information). Therefore, they did not select a spill method and instead input the free flow rating curve (maximum total outflow vs. pool elevation) in the Max Release slot on storage reservoirs and in the Max Flow Through Turbines slot on power reservoirs. RiverWare will solve normally when configured in this fashion. Instead of using spillways and regulated spill structures, all outflow is sent through the Release slot (on storage reservoirs) or the Turbine Release slot (on power reservoirs). While this will not affect the reservoir calculations, it is misleading because the Turbine Release and Release slots will show outflow for both the release and spill structures. This would not work if the reservoir is being used to compute power. In that case, a spill method should be selected. The sum of the turbine release capacity and spill capacities should equal the free flow rating curve.

2.2.4.3 Order of Spillway and Release Structures Used in RiverWare

If more than one spillway or release structure is being modeled on a given reservoir, RiverWare sends the reservoir outflow through these structures in a specific order. If there is an uncontrolled spillway, the spill over that structure is computed based on the average pool elevation over the timestep. Any remaining outflow is then sent through the turbines, if it is a power reservoir, or through the Release slot if it is a storage reservoir (the maximum release information is held in the Max Release slot). If there is any remaining outflow, it is sent through the Regulated Spill slot or the Bypass spill slot depending on the spill method selected. In general, RiverWare assumes that all outflow should go through the turbines (or the Release slot on storage reservoirs) and any excess is sent through the regulated spill structures.

2.2.5 Inflows

There are many slots on the reservoirs that can represent sources and sinks to the mass balance. This section describes the recommended structure for the inflows to the reservoir.

2.2.5.1 Reservoir Inflow and Mass Balance Data

The Inflow slot represents the inflow to the reservoir from upstream (routed flows from upstream, not uncontrolled area flows). Therefore, the Inflow slot on all reservoirs should be linked to the Outflow slot of the upstream object unless the reservoir is a headwater reservoir. If it is a headwater reservoir, a flow rate of zero should be input for all timesteps (including values through the last dispatch timestep) in the Inflow slot (the reservoir will still receive inflow from the Hydrologic Inflow Forecast slot).

Since the inflow to each reservoir will be known (either propagated from an upstream object or input to zero), one other piece of data must be input for each timestep to allow the reservoir to mass balance. An initial value for either storage or pool elevation must be input and then values must be input at each timestep for either pool elevation, storage or outflow. For testing purposes, and to ensure proper mass balancing, various combinations of inflow, outflow, storage and pool elevations can be input and evaluated.

2.2.5.2 Forecast Inflow - Uncontrolled Area Flow

The reservoir objects (and control points) are used to bring uncontrolled area flows into the system. The USACE-SWD data for uncontrolled area flows is the cumulative, uncontrolled area flow from the upstream reservoir to the control point, adjusted for routing. As a result, it cannot be input directly into the Local Inflow slot or the flows would be multiply counted. Also, the uncontrolled flows must be forecasted to provide an estimate throughout the forecast period. Specific methods must be selected to deal with these two unique features. This is described in detail **HERE (Section 2.8)**.

2.2.6 Seepage

Seepage through the face of the dam is modeled by selecting the **Input Seepage** of **Single Seepage Value** method in the Seepage category. These methods allows the user to input a timeseries of seepage values or a constant value. The Reservoir.Seepage slot should then be linked to the downstream object's Local Inflow slot. The screenshot in Figure 3 shows the correct linking structure.

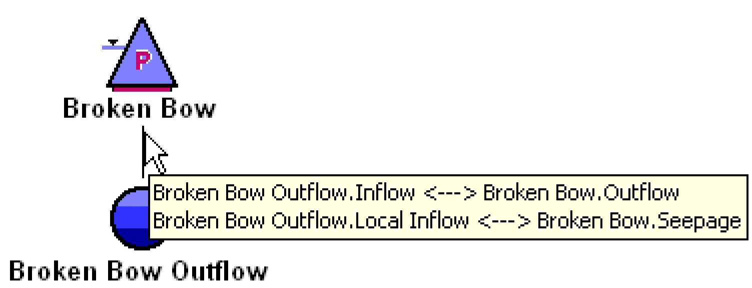


Figure 3. Linking Seepage

2.2.7 Other possible user methods

There are several other user method categories available on the reservoir that are not discussed in this document. Decide what additional physical processes need to be modeled (if any) and then select the appropriate methods and input the appropriate data. Possibilities include evaporation and precipitation, or bank storage. The object documentation **HERE (Objects.pdf, Section 24.1)** describes the various methods and the slots associated with each.

2.3 Reach Objects

The Reach object is used to route water through the network. Also, water diversions from the river must come out of the Reach object. The Inflow slot should be linked upstream and the Outflow slot should be linked downstream. The user should consult the reach documentation **HERE (Objects.pdf, Section 22.1)** for information about any user methods not discussed in this document.

2.3.1 Routing

The **Step Response** routing method **HERE (Objects.pdf, Section 22.1.1.5)** is typically used by the USACE-SWD. This is the same routing algorithm used in the FloodControl predefined function in RiverWare. If the user chooses a different routing method, then the final releases will be routed with a different method than that used in the algorithm to determine the flood control releases.

If the **Step Response** method is selected, input the number of routing coefficients in the Num of Coeff slot on the reach object. The routing coefficients are input in the Lag Coeff slot. These coefficients are the INCREMENTAL routing coefficients that apply to that reach specifically (as opposed to the direct routing coefficients used on the control points for the computation of flood control releases).

Sometimes, the routing coefficients are not fixed, but are dependent on the flow in the reach, particularly during high flow events. In this situation, the **Variable Step Response** routing method (**HERE (Objects.pdf, Section 22.1.1.6)** should be used. The entire alternative routing algorithm is described in detail **HERE (Section 2.10)**.

2.3.2 Solve Downstream Only

On any reach in the model that does not have routing (i.e. the **No Routing** method is selected), the user should select the **No Local Inflow, Solve Outflow** (described **HERE (Objects.pdf, Section 22.1.3.7)**) method in the **Local Inflow and Solution Direction** category. This method forces the reach to always solve in the downstream direction. This eliminates the possibility of unintentional upstream solving as a result of complicated rule priorities.

2.3.3 Transit Losses

In certain basins, the USACE-SWD simulates the loss of water released from a reservoir to meet downstream demands. In the **GainLoss** category, the **Base Plus Fractional Loss** method (**HERE (Objects.pdf, Section 22.1.10.6)**) models loss as:

- a base loss, and
- a loss that is a fraction of flow above that base loss, and
- the total loss cannot exceed a maximum flow rate

In addition, the base loss and fraction above that base can be entered seasonally for future extensibility. Select this method or another method in this category to model transit losses.

2.4 Confluence Objects

The confluence object brings together flows from two objects. Select the **Solve Downstream Only** method (described **HERE (Objects.pdf, Section 8.1.1.2)**) in the **Confluence Solution Direction** category. This method forces the confluence to always solve in the downstream direction. This eliminates the possibility of unintentional upstream solving as a result of complicated rule priorities.

2.5 Control Point Objects

The control point models a location on a river where certain flow requirements (maximum and minimums) must be maintained. 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. They also represent the location for low flow demands and additions of uncontrolled local flows.

2.5.1 Local Inflows / Uncontrolled Area Flow

Control points (and reservoirs) are used to bring uncontrolled area flows into the system. The USACE-SWD data for uncontrolled area flows is the cumulative, uncontrolled area flow from the upstream reservoir to the control point, adjusted for routing. As a result, it cannot be input directly into the Local Inflow slot or the flows would be multiply counted. Also, the uncontrolled flows must be forecasted to provide an estimate throughout the forecast period. Specific methods must be selected to deal with these two unique aspects. This is described in detail **HERE (Section 2.8)**.

Control Points located at the outflow of a reservoir (or those that don't have local inflows) should not use these methods; they should keep the None method selected in the Local Inflow category.

2.6 Water User Objects

The water user diverts water from either a reach or reservoir, uses a certain amount, and returns the rest. The USACE-SWD uses Water User objects to simulate reservoir and reach diversions in their models. In the simulation/calibration model, the user should only configure water users to make direct from reach diversions. Any diversions from the reservoirs should be input directly on the reservoir's Diversion slot. When the policies are implemented to determine the reservoir diversions, there are additional methods and linking structure that must be created. These are described **HERE (Section 3.8)**.

Typically, some water users remove water from the system but do not return any water. Other Water Users are used to divert from one reservoir or reach and pump it into other reservoirs. These Water Users "return" all of the diverted flow to the other reservoir. On each water user, the user should select the Fraction Return Flow method in the **Return Flow** category. Then, in the Fraction Return Flow Input category select either:

- Input Fraction: The user provides a series of Fractional Return Flows. **HERE (Objects.pdf, Section 27.1.5.1)**

- Zero Fraction: The return flow is set to zero and all water is assumed to be depleted, **HERE (Objects.pdf, Section 27.1.5.2)**
- Periodic Fraction: A periodic slot will represent the fractional return flow. The USACE-SWD can input a 1.0 to represent that all water diverted is to be returned. **HERE (Objects.pdf, Section 27.1.5.3)**

2.7 Data Objects and Custom Slots

Data objects are useful for storing data that is not directly applicable to a simulation object. For example, a data object can be used to hold system data. Custom slots can be created on data objects or on simulation objects. For example. custom slots can be created on the reservoir to hold historical reservoir data for easy comparison with the RiverWare results. Also, custom expression slots and statistical table slots can be used for post processing and analysis of the data. Data objects are described in more detail **HERE** and the some specific USACE-SWD analysis tools are described **HERE (Section 5.2)**.

2.8 Forecasting Incremental Local Inflows from Cumulative Flows

In USACE-SWD basins, streamflow data represents the cumulative inflow to the river up to that point. These cumulative inflows cannot be routed downstream in RiverWare since the next downstream Control Point also has cumulative inflows. However, routing inflows in RiverWare is useful so that diversions can be made from Reaches and so that the Reaches will be routing the correct flows. In order to address this problem, methods were developed to use the cumulative inflows at Control Points and Reservoirs, and calculate the Incremental Flows. The methods calculate the forecasted incremental local inflows at each control point and reservoir within a subbasin given the cumulative local inflows. These methods are described in this section and in the objects specific help.

2.8.1 User Implementation

Users will first need to set up a computational subbasin for each group of control points and reservoirs that contain cumulative local inflow data. The computational subbasin(s) must include all control points and reservoirs which contain the cumulative local inflow data as well as all the intervening reaches and confluences. Each control point and reach should only be included in one reservoir's subbasin. For example, in the sample screenshot, Figure 4, there is a subbasin that contains Clayton, Clayton_Antlers, Antlers, Antlers_Diver_Reach, Antlers_Hugo, and Hugo. The subbasin does not go upstream any further because Sardis Outflow and Tuskahoma Dam Site either have no locals or the locals are already included in Outflow.

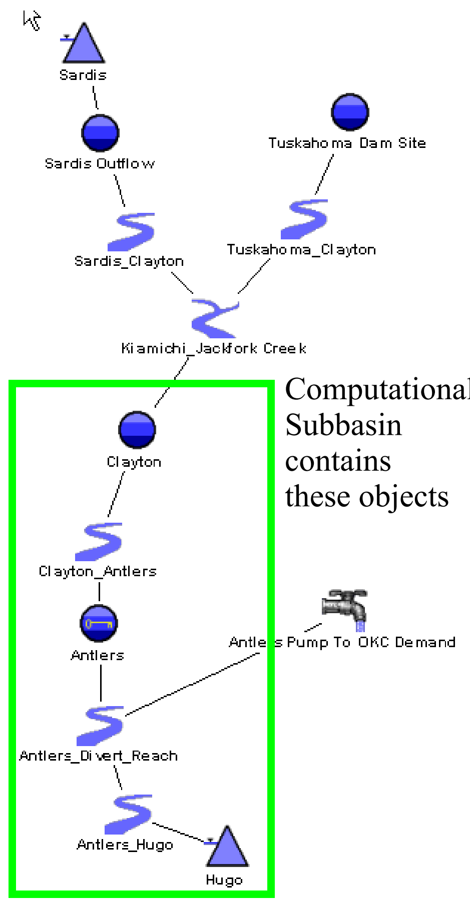


Figure 4. Sample computational subbasin

For each computational subbasin, select the **Compute Forecast Period Incremental Local Inflows** method from the **Compute Incremental Local Inflows** category. This enables the **Reservoir Boundary for Incrementals** category. Because cumulative local inflow data is only cumulative until a reservoir is reached, select the **Stop at Reservoirs** method from the **Reservoir Boundary for Incrementals** category.

Additionally, all headwater reservoirs and those with reservoirs with no upstream objects that have cumulative inflows should be placed in a single computational subbasin. Then on the subbasin, select the **Compute Forecast Period Incremental Local Inflows** method from the **Compute Incremental Local Inflows** category. This enables the **Reservoir Boundary for Incrementals** category. Select the **Reservoirs Only** method from the **Reservoir Boundary for Incrementals** category (**HERE (Objects.pdf, Section 7.1.23.4)**). This method allows the user 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.

Note: Make sure to enter values after the run so that post-run dispatching can occur. That is, you only need to input Cumulative Hydrologic/Local Inflows through the run end plus the Period of Perfect Knowledge on each inflow location. Click **HERE (Section 2.10.4)** for more information.

For each control point and reservoir in each computational subbasin several new methods must be selected. Method selection on multiple objects can be accomplished efficiently through the Multiple Object Method Selector. The Multiple Object Method Selector is enabled by selecting Workspace ➥

Objects ➥ Select Methods on Objects.

In RiverWare, local inflows are called “Local Inflows” on Control Points and “Hydrologic Inflows” on Reservoirs. In this discussion, we will use the terminology Local Inflows exclusively; please substitute Hydrologic Inflows anywhere a reservoir is referenced.

For each control point and reservoir in an incremental calculation subbasin, users should perform the following:

- Select **Forecast Local Inflows** method (in the **Local Inflow** category) to specify that forecasting should be performed.
- Specify how the forecasting should be performed. Typically, for the USACE-SWD, the **Geometric Recession** method should be selected in the **Generate Forecast Inflows** category.
- Select the **Compute Forecast Period Incremental Local Inflows** method from the **Compute Incremental Local Inflows on Subbasin** category (**Compute Incremental Hydrologic Inflows on Subbasin** category on reservoirs).
- On the control points, the **Locals Included In Outflow** method should be selected from the **Include Locals in Outflow** category.

With this setup, the user then inputs the cumulative flows into the Cumulative Local Inflow slot (on control points) or the Cumulative Hydrologic Inflow slot on the reservoirs.

2.8.2 Methods

Each of the above listed methods is described in the object specific documentation:

Storage and Level Power Reservoirs:

Category	Method	Storage Res Link	Level Power Res Links
Hydrologic Inflow	**Forecast Hydrologic Inflows**	HERE (Objects.pdf, Section 24.1.9.5)	HERE (Objects.pdf, Section 17.1.15.5)
Generate Forecast Hydrology	**Geometric Recession**	HERE (Objects.pdf, Section 24.1.10.2)	HERE (Objects.pdf, Section 17.1.16.2)
Compute Incremental Hydrologic Inflows on Subbasin	**Forecast Period**	HERE (Objects.pdf, Section 24.1.11.3)	HERE (Objects.pdf, Section 17.1.17.3)

Control Point:

Category	Method	Control Point Link
Local Inflow	Forecast Local Inflows	HERE (Objects.pdf, Section 9.1.1.3)
Generate Local Inflows	Geometric Recession	HERE (Objects.pdf, Section 9.1.2.2)
Compute Incremental Local Inflows on Subbasin	Compute Forecast Period Incremental Local Inflows	HERE (Objects.pdf, Section 9.1.6.3)

Computational Subbasin:)

Category	Method	Comp. Subbasin Link
Calculated Incremental Local Inflows	Compute Forecast Period Incremental Inflows	HERE (Objects.pdf, Section 7.1.22.3)
Reservoir Boundary for Incrementals	Any method	HERE (Objects.pdf, Section 7.1.23)

2.9 Initialization for Routing methods

In routing reaches, the user must input enough data before the start of the run (pre-simulation) to allow the reaches to solve for the Outflow at the first (and possibly subsequent) timestep. The necessary slots and number of pre-simulation values required to ensure solution of the entire system at the start timestep depends on the number, type, linkages, method selection and parameterization of objects in the system.

In most circumstances, the modeler identifies the pre-simulation data requirements through knowledge of the system or trial and error. These data are then specified (manually or via input dmi's) to provide the model with required and reasonable data. This section describes an approach to automatically initializing flow slots at pre-simulation timesteps so routing can simulate successfully at the first computational timestep.

2.9.1 Slots requiring pre-simulation values

The following section describes the slots used in USACE-SWD models requiring pre-simulation data:

1. Outflow from each reservoir to initialize inflows to downstream routing reaches. Note, reservoirs never dispatch before the start of the simulation and as a result, reservoirs need neither Inflow nor Hydrologic Inflows, before the start of the run. Of course, reservoirs still require 1 initial value for pool elevation or storage. Also, certain methods require a single value for the initial timestep. For example, the tailwater methods require an initial Tailwater Elevation. These single initial values will still need to be input by the user.

2. Seepage from each reservoir, when in use and linked. Typically, the reservoir Seepage slot is linked to the outflow Control Point's Local Inflow slot. When non-zero, Seepage is typically a small value. But the Seepage needs to be initialized to provide reasonable flows downstream on the first timestep.

3. Flow into the upstream-most non-reservoir object of each tributary, typically a control point inflow or possibly a reach inflow. In the case of many models, these are all control point objects (often called "Dam sites").

In the USACE-SWD models, secondary flow paths have been created to allow Low Flow Releases (click **HERE (Section 3.7)**) to be diverted around the routing reaches as shown in the following screenshot, Figure 5. Because this modeling approach requires a "No Routing" reach and a water user, additional pre-simulation values are required to solve these two reaches at the first timestep:

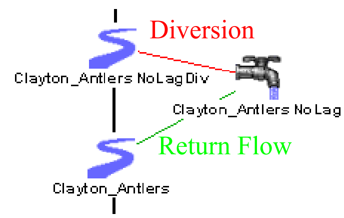


Figure 5. Linking structure for secondary flow paths

4. Diversion on reaches diverting to water users. In addition, any direct-from-reach diversions will also need the Diversion slot initialized.

5. Return Flow on reaches when linked to water users.

Technical note: although Diversion and Return Flow are not typically required to solve a reach, if these two slots are linked but unknown, the reach will dispatch but not solve.

In the above screenshot, the two reaches are linked to the water user (Reach.Diversion to Water User.Diversion, Reach.Return Flow to Water User.Return Flow). As a result, it is only necessary to set one side of the link and it makes no difference which side is set. Therefore, it would be identical to initialize the Water User's Diversion and Return Flow slots.

6. Local Inflow on reaches, when instantiated.

2.9.2 Pre-simulation values

Currently in USACE-SWD models, all pre-simulation values are set to zero. On the Local Inflow, Diversion, and Return Flow slots, this value may be reasonable. But zero reservoir outflow does not seem reasonable as this value will propagate downstream. Therefore, the above slots can be initialized using one of the following methods:

1. Assign zero to all pre-simulation timesteps.

2. Assign the user-input value at the initial timestep to all pre-simulation timesteps on that slot.

2.9.3 Required Number of Pre-Simulation timesteps

For each object, the method determines the number of required pre-simulation values by visiting each downstream reach object and looking at the routing characteristics. The search is stopped when a reservoir or other object that does not dispatch before the first timestep is met or the downstream-most point in the basin is found. Objects that do not dispatch before the timestep are: Reservoir (all types), Pipeline, Inline Pump and Bifurcation. The upstream to downstream search will stop at any of these points.

Note that even though the Gage does not dispatch before the start of the timestep, its inflow and outflow are linked, allowing any upstream values to propagate downstream.

Reservoirs of all types, Pipelines, Inline Pumps and Bifurcations will not calculate a mass balance for timesteps before the begin of the run, so reaches downstream of these points must be initialized as if they were at the top of their basins. For example, flows above a reservoir will not affect the outflows specified for the reservoir because the reservoir cannot solve prior to the start timestep.

To ensure that this algorithm produces the same results for model files without the initialization method, every object is checked to see that it is a member in a subbasin with an initialization method. If the object is not a member, the number of lags calculated will be zero, and the results of the first dispatch timestep will be the same as without these methods selected.

The number of required pre-simulation values will be determined as follows:

Visit each reach in the flow path and sum the number of lag coefficients. For each object, the required number of pre-simulation timesteps is:

$$(\text{Number of Lag Coefficients} - 1) \qquad \textbf{(EQ 1)}$$

Note, this approach assumes you can determine the maximum number of coefficients *a priory*. Therefore, it only works for the No Routing, Step Response, Variable Step Response, Impulse Response, Time Lag, or Variable Time Lag methods.

When diagnostics are turned on, Reaches and Aggregate reaches report the total number of downstream timesteps required in order to solve the most downstream object at the first timestep. Aggregate Reaches, like other objects, report the total number of timesteps required, but because they contain reaches, the contained reaches also report the number of timesteps required.

2.9.4 Methods

The above methods are described in the computational subbasin help, **HERE (Objects.pdf, Section 7.1.25)**.

2.9.5 Using Initialization methods

To apply the initialization functionality to an existing USACE-SWD model, three steps are required: create a subbasin, define initial values as necessary, and disable existing initialization rules. This section will identify and walk through these steps.

2.9.5.1 Create an Initialization Subbasin

The Initialization methods reside on a Computational Subbasin. While several computational subbasins exist in USACE-SWD models, it is recommended that a new subbasin be created to ensure that all simulation objects be considered for initialization; not just those listed in an existing subbasin.

- Open the **Workspace ➥ Edit Subbasins** dialog.

- Select the **Subbasin ➥ Append Typed Subbasin ➥ Computational Subbasin**.
- Click on the new subbasin to rename it, then select **Subbasin ➥ Invoke Member Selector**.
- Use the Selector to select all simulation objects (excluding data objects) as members of the subbasin.
- Close the Subbasin Manager dialog and return to the workspace, select **Workspace ➥ Open Computational Subbasin ➥ <Name of the New Subbasin>**. Shift to the methods tab and select the desired initialization method from the **Initialize Flow Slots for Routing** category. For simplest initialization, use **Backcast Zeros.** Otherwise, select **Backcast Initial Value**.

If further flexibility is required, the user may create separate subbasins to initialize objects differently. For example, choose to specify all reservoirs as part of a Computational Subbasin that Backcasts the Initial Value. All the other objects should be part of a Computational Subbasin that would Backcast Zeros. This approach is possible because all of the objects in the model should be in a subbasin to find the correct downstream lag, they don't necessarily have to be in the same subbasin.

2.9.5.2 Define initial values as necessary

In past USACE-SWD models, slots requiring initialization values were set with rules and/or were set manually. If desired, these user-input values may now be removed, resulting in the model relying more fully on the Initialization methods. Remember that these user input values or any "extra" values *do not* interfere with proper execution of the Initialization methods, but may be unnecessary. While there are several means to accomplish removing the initial values, during development and testing CADSWES staff found one method to be relatively efficient:

- Create a new SCT and add all existing slots to it.
- Under **Config ➥ General**, extend the display period by setting the Pre-simulation Timesteps value to a high number.
- Scan through the slots focusing at or near the Initial Timestep for those shaded as User Input (gray). When a slot is found with user input, remove those values as appropriate. If you have selected to **Backcast Initial values**, make sure to leave one input value on the initial timestep.

Remember that certain slots require user input at the Initial Timestep, such as Reservoir Storage or Pool Elevation.

2.9.5.3 Deactivate or Remove Initialization Rules

If previous initialization efforts included usage of rules to initialize diversions, return flows or local inflows, these rules should be removed or deactivated. If the chosen method is **Backcast Zeros**, the model should run.

2.9.5.4 Other suggested changes

Change no-lag reaches to use the **No Local Inflow, Solve Outflow** (**HERE (Objects.pdf, Section 22.1.3.7)**) method to avoid requiring the Local Inflow to have a initial value.

Diagnostics are available for the initialization methods under the **Dispatch Management, SimObj** category. Make sure to select the computational subbasin in the object filtering section.

2.10 Alternative Routing Coefficients methods

In the USACE-SWD flood control routing methods, releases are routed using the Step Response routing method. During extreme large events with overbank flow, the channel routing coefficients are insufficiently accurate to model the routing of water down the reach, i.e., large flows that occupy the overbanks have different travel times. This section outlines the required simulation functionality and proposes a design to add the capability to RiverWare to provide for alternative routing coefficients during high flows, both in the reach simulation and in the flood control release algorithm.

The USACE-SWD typically uses the Step Response Routing method to route water in RiverWare simulations. There are two separate sets of routing coefficients used, one set for each control point and one set for each reach. The control point routing coefficients are used in the flood control algorithm's release determination while the reach routing coefficients are used in the simulation to route water. These two routing methods are described below.

One set of routing coefficients for each upstream reservoir resides on the control point Routing Coefficients slot. These coefficients are used only in the flood control algorithm to estimate the effect reservoir releases would have on downstream flooding. In the flood control algorithm water is routed from each upstream reservoir directly to the control point (i.e. reservoir outflow routed to control point) using the following equation:

$$\begin{aligned} \text{ControlPointFlow}_t &= C_0\text{ResOutflow}_t + C_1\text{ResOutflow}_{t-1} + \\ &\ldots + C_{ncoeff-1}\text{ResOutflow}_{t-ncoeff-1} + C_{ncoeff}\text{ResOutflow}_{t-ncoeff} \end{aligned} \quad \textbf{(EQ 2)}$$

The estimated flow at the control point is the sum of the contribution from each upstream reservoir. This equation is only used in the routing in the flood control calculation (i.e. rule function evaluation) and does not actually route water to the control point. These calculations are used by the rule to set reservoir outflows.

The reach is used to route water in the simulation after reservoir outflows have been set. On the reach, the following equation is used to incrementally route water in the reach (i.e. route Inflow plus Local Inflows to Outflow) during the simulation. Step Response routing uses the equation:

$$\begin{aligned} \text{Outflow}_t &= C_0\text{Inflow}_t + C_1\text{Inflow}_{t-1} + \\ &\ldots + C_{ncoeff-1}\text{Inflow}_{t-ncoeff-1} + C_{ncoeff}\text{Inflow}_{t-ncoeff} + \text{TotalGainLoss} \end{aligned} \quad \textbf{(EQ 3)}$$

This equation is used to calculate the current and future outflows (from t = current to t = number of coeffs). All of the C values are calculated outside of the model and for each set of coefficients, the C values sum to 1.0.

To simulate large flows accurately, it is necessary to use alternative routing coefficients only at high flows for both the flood control calculation and the reach simulation. These routing coefficients come into effect when flows are above thresholds. These coefficients are used when the flow is above a threshold in both the flood control algorithm and the reach routing.

Note: It is not necessary to use Step Response Routing on the reaches for simulation/dispatching. You can use any routing method you want. Of course the routing will not match the linear routing used in the flood control algorithm. If you do use a different routing method on the reaches, you will need to still specify routing coefficients on the Control Points. If you wish to use flow based routing coefficients, you will need to select the Variable Step Coefficients method in the Alternative Routing on Subbasin category. This is described below.

2.10.1 User Implementation

To implement the above approach, various methods must be selected.

In the simulation/calibration model, you have two choices:

- Select the **Variable Step Response** routing method on the reach. This method adds the alternative coefficients table slot.
- If you wish to not use the **Variable Step Response** method for simulation/dispatching, select a different routing method, like Modified Puls. But, you must then choose **Variable Step Coefficients** in the **Alternative Routing on Subbasin** category.

Then, populate the **Variable Lag Coefficients** table slot with data.

When Flood Control is implemented, a computational subbasin must have the correct method selections. Generally, the same subbasin used for Flood Control (**HERE (Section 3.6)**) should be used for this purpose. This subbasin will contain most, if not all objects in the basin. On this subbasin, select the **Compute Aggregate Coefficients** method in the **Control Point Variable Routing Coefficients** category.

Additionally, on each control point downstream of a reach with alternative coefficients, the user must select the **Compute Aggregate Coefficients** method or **Compute Aggregate Coeffs every Timestep** method in the **Variable Routing Coefficients** category. No additional data entry is required on the control point for this method (although the standard Routing Coefficients table are typically input on the control point for the normal flow conditions).

2.10.2 Methods

To enable this approach, the user must select methods on three objects:

Object	Category	Method	Link
Reach	**Routing Method**	**Variable Step Response**	**HERE (Objects.pdf, Section 22.1.1.6)**
	Alternative Routing on Subbasin	**Variable Step Coefficients**	**HERE (Objects.pdf, Section 22.1.21.2)**

Object	Category	Method	Link
Control Point	**Variable Routing Coefficients**	**Compute Aggregate Coefficients**	HERE (Objects.pdf, Section 9.1.19.2)
		Compute Aggregate Coeffs every Timestep	HERE (Objects.pdf, Section 9.1.19.3)
Computational Subbasin	**Control Point Variable Routing Coefficients**	**Compute Aggregate Coefficients**	HERE (Objects.pdf, Section 7.1.24.2)

2.10.3 Alternative Routing and Flood Control

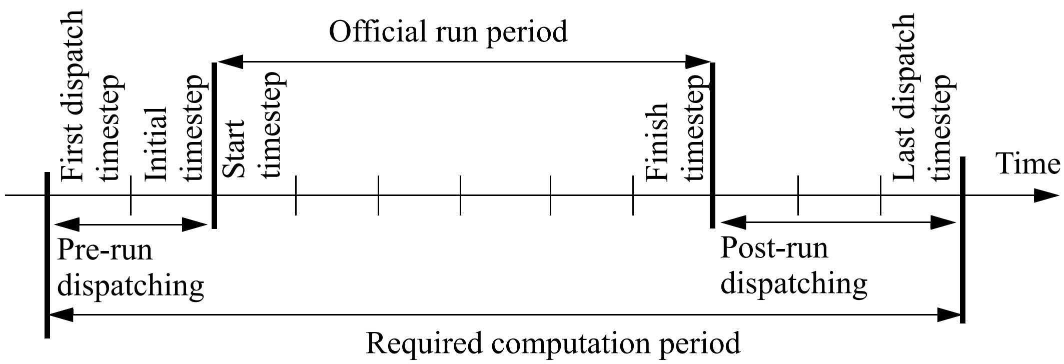
The flood control algorithm accesses the correct set of routing coefficients based on the method selection. On a given control point,

- If the **Compute Aggregate Coefficients** method is selected, the calculated **Computed Routing Coefficients** will be used. These represent the coefficients determined by the subbasin at the beginning of the timestep. If they were not re-calculated (flows did not go above the first threshold), they will contain the same values as the Routing Coefficients slot.
- If the **Compute Aggregate Coeffs every Timestep** method is selected, the calculated **Computed Routing Coefficients** will be used. These represent the coefficients determined by the subbasin at the beginning of every timestep.
- If the **None** method is selected on the control point, the existing Routing Coefficients slot will be used.

Once the set of coefficients are determined, the flood control algorithm proceeds as usual. The algorithm will use the coefficients throughout flood control execution and throughout the forecast period.

2.10.4 Post-run Dispatching

Forecasting and basin lag times result in situations where the last timesteps of the run may not solve correctly if the objects have not solved past the end of the run. To allow objects to solve past the end of the run, you must configure **post-run dispatching**. The figure below shows a time line diagram when post-run dispatching will occur.



To configure post-run dispatching, set the number of post-run dispatch timesteps equal to the forecast period as described **HERE (RunControl.pdf, Number of Post-Run Dispatch Timesteps)**. For dispatching to occur after the end of the run, certain data must be specified. Following is a description of the required data for post run dispatching

Local Inflow Forecasting Algorithms: The forecasting and disaggregation of cumulative local inflows is done at the beginning of each timestep. The forecasting will be executed at Begin Timestep on the last timestep, but will not occur after that. That is, even though the objects dispatch past the end of the run, no additional forecasting is required. But, local inflow data is required after the end of the run through the Period of Perfect Knowledge. That is, the user only needs to input Cumulative Local Inflows through the run end plus the Period of Perfect Knowledge on each inflow location.

Data that is filled for each object: Following is data that will be automatically filled in at beginning of run and there is not a valid value. No user interaction is required.

- Reservoir.Diversion
- Reservoir.ReturnFlow
- Reservoir.Precipitation Rate

Data that will have to be input: Following is data that is typically input by the user (input or using a DMI). These will need to be input for post-run timesteps as well. DMIs or rules (RBS or Init) can be configured to quickly set this data.

- Headwater inflows: Gage.Inflow, Control Point.Inflow, Confluence.Inflow 1 or 2 or Reservoir.Inflow, Reach.Inflow. These are typically set to zero.
- Reservoir.Evaporation Rate or Evaporation (defaults to zero if not input)
- Reservoir.Cumulative Hydrologic Inflow and Control Point.Cumulative Local Inflow (or Deterministic Incremental Hydrologic or Local Inflow) See section above for data requirements.

Data that is currently input but can be set by alternative method selections: Following is data that is may be input, but with different method selections, the data can be automatically set for all timesteps, including post-run.

- WaterUser.Fractional Return Flow: Select “Periodic Fraction” method and input a value. Make sure there are no other input values on these slots.
- Reservoir.Seepage: Select “Single Value Seepage”. Otherwise, seepage defaults to zero if not input.

USACE-SWD Methods - Policies

3. Implementing Operating Policies using Rulebased Simulation

Once the basic model structure has been developed and calibrated using the Simulation controller, the operating policies can then be implemented using rules. The rules will compute the reservoir outflows and diversions to water users based on the state of the system and the operating policies. Described in this section is an overview of configuring the rulebased simulation model followed by a description of each USACE-SWD policy: surcharge, regulation discharge, flood control, low flow releases, reservoir diversions, hydropower releases, and fish releases. Described for each policy are the additional method selections and data needs. Then the implementation in RPL is described along with algorithm itself.

3.1 Overview of use of Rulebased Simulation

The USACE-SWD models includes policy implemented as rules. Therefore, the model must use the Rulebased Simulation controller. The Rulebased Simulation controller is selected in the Run Control dialog.

A Rulebased Simulation model is an under determined model in terms of user input values. A set of prioritized rules, which implement basin policy, is used to look at the state of the system, execute policy logic, and set slot values to provide enough information for the model to solve.

In the calibration model discussed above, storage values may have been input for each timestep on every reservoir. The user may have chosen to input outflows or pool elevations instead. Regardless, in the rulebased model, this data must be removed because rules will be used to determine the reservoir outflows. However, this data will be useful for testing the various flood control algorithms so it should be stored on a data object. So for each reservoir object, copy input storage values into a custom slot and then clear those values from the reservoir object (every reservoir still needs an initial storage or pool elevation value so don’t remove those). Once these storages are removed, the model is under determined and will not solve. The flood control rules, whose implementation is discussed below, will be used to provide the necessary data for the model to solve.

3.2 Ruleset Overview

Rules are used to implement policy and invoke the flood control algorithms taken from the SUPER logic. Rules execute in the order specified by the user and are numbered according to priority. Rule #1 is the highest priority rule while the rule with the highest number (depends on the number of rules in the ruleset) is the lowest priority rule. In general, the lowest priority rules should execute first and represent the ideal operating conditions while the highest priority rules should execute last and represent special conditions or constraints that must be upheld. The idea behind rulesets meant to reproduce the SUPER logic is a bit different. Rulesets that implement SUPER flood control logic are designed to make mandatory releases first, then the releases may be increased as space allows. Therefore, the surcharge release rules execute first, then the rules to compute the regulation discharge and empty space, and finally the rules to compute the flood control releases. After the flood control model is working properly, additional higher priority rules could be added to account for other types of releases (for fish flows, power generation, irrigation demands, etc.).

3.2.1 Creating a new ruleset

A new ruleset is created by selecting **Policy ➥ Ruleset ➥ New** in the main RiverWare menu. A full description of the ruleset editor user interface is described **HERE (RPLUserInterface.pdf, Section 1)**.

3.2.2 Agenda order

To follow the guidelines presented in this document, the ruleset should execute the rules in order of lowest priority rules first and the highest priority rules last, that is rule *n* first, rule 1 last. This is the default structure but the following procedure can be used to verify this: Click **View ➥ Show Advanced Properties** menu in the RPL set editor. In the **Agenda Order** field, select the toggle button for...,3,2,1.

3.2.3 Order of policies

In a rulebased simulation run, the run controller alternates between rule execution and object simulation. For a given controller timestep, all rules start out on the agenda. Rules are executed in reverse priority order until a rule succeeds and sets a slot in the model. Then control shifts to simulation and objects dispatch given the new information from the successful rule. When simulation is finished and there are no objects left to solve, control shifts back to the rule agenda and rules continue to execute until one is successful. Then control shifts back to simulation to simulate the affects of the rule. This process continues until there are no rules left on the agenda and no objects to solve. Then the controller moves to the next timestep and the process begins again. Two important points to remember are: 1) after a rule executes and is taken off the agenda it can go back on the agenda if one of its dependencies changes and 2) objects can solve at timesteps other than the current controller timestep if they have enough information to do so.

The following table shows the priority and execution order to be used by most USACE-SWD models. The rules will execute from lowest priority to highest (8 to 1) as shown by the execution order. Following is a description and the rationale (sorted by execution order)

.

RPL Priority Order	Execution Order	Policy
1	8	Fish Releases (QFish)
2	7	Hydropower Release
3	6	Reservoir Diversions
4	5	Low Flow Releases
5	4	Flood Control
6	3	Regulation Discharge
7	2	Minimum Flood Control Releases (Optional)
8	1	Surcharge Release

0. The forecast of incremental inflows from cumulative methods (**HERE (Section 2.8)**) executes for each control point and reservoir object at the start of each timestep. This results in a new inflow forecast for each object. The computed inflow forecast sits on the Hydrologic Inflow Forecast slot on reservoir objects and on the Local Inflow slot on the control point objects. This is listed as step zero because it happens at the beginning of the timestep (and at priority 0) before any rules execute. Also at beginning of the run, the selected method in the Conditional Operating Level category (**HERE (Objects.pdf, Section 22.1.20)** will be executed. This could alter which operating level table will be used.

1. The lowest priority surcharge release rule is executed and it sets the surcharge release flag on the outflow slot of a reservoir object. That reservoir object then dispatch with the solveMB_givenInflowOutflow dispatch method. While dispatching, the reservoir computes a surcharge release forecast. These values will be set on the Surcharge Release slot and on the Outflow slot. The reservoir object will then solve for storage and pool elevation over the forecast period using the new outflow values. When the reservoir is finished dispatching, the forecast outflows will be routed downstream to the next reservoir object. This is done by dispatching the reach and control point objects as they receive inflow values from upstream (their inflow values are the routed releases from the upstream reservoir). When all objects are finished dispatching, the next highest priority rule executes and sets the surcharge release flag on another reservoir object. That reservoir then dispatches and computes its surcharge release forecast. The forecasted outflow values are then routed downstream to the next reservoir. This process repeats until all surcharge release rules have executed, all reservoirs have computed a surcharge release forecast, and the surcharge releases have been routed downstream.

Note: The surcharge policy initializes the reservoir Outflow to zero if no surcharge release is necessary.

2. Optionally, minimum flood control releases can be made after the surcharge operation. These releases are considered minimums because they are made regardless of what the flood control logic computes. Also, the releases occupy downstream channel space so they should be made before the

Regulation Discharge. It is up to the user to write the rule logic for when these should be made, but the rule should set the

- Reservoir.Flood Control Minimum Release
- Reservoir.Outflow
- Reservoir.Bypass or Regulated Spill (if this water should not go through the turbines or release structure)

Click **HERE (Section 3.4)** for more information.

3. The regulation discharge rule executes and sets the regulation discharge flag on the Reg Discharge Calculation slot on each control point object. This triggers each control point to dispatch. The control points compute the regulation discharge and empty space hydrographs over the forecast period. Because each control point has already computed its inflow forecast and the surcharge releases have already been routed from upstream reservoirs, the control point knows how much water is already occupying the channel at its location and can compute the empty space accordingly.

4. The flood control rule executes and calls the FloodControl predefined function. For all reservoirs in the computational subbasin and for all timesteps in the forecast period, the flood control releases are computed. The FloodControl function returns the final flood control releases and the final outflows (flood control release plus surcharge release) for the current timestep. The flood control rule sets the Outflow and Flood Control Release slots. This triggers each reservoir object to re-dispatch with the solveMB_givenInflowOutflow method. The reservoirs compute their final storage and pool elevation values and the outflows are routed downstream.

5. The Low Flow Release rules execute next, releasing additional flows to meet downstream requirements. Flood control and low flow releases are normally thought of as mutually exclusive for a particular reservoir in that when a flood control release is necessary, minimum flows downstream are satisfied and additional low flow releases are not necessary. However, it is possible for some reservoirs to be in flood control while others are required to make low flow releases. Also, in SUPER (hence in the RiverWare model), demands for diversions out of the rivers are satisfied by specifying "low flow" requirements at control points upstream of the diversion points. Because these low flows are not restricted to environmental low flows, but also include diversion demands, it is possible that flood control releases are not adequate to satisfy these demands and additional releases are needed from the conservation pool. Hence the Low Flow Release rules execute regardless of whether flood control releases have been executed.

6. The Reservoir Diversion rules execute next, diverting water out of reservoirs directly to water users or to other reservoirs. This happens after the Low Flow rule executes, guaranteeing that the low flows will be satisfied at the possible expense of direct reservoir diversions. This is the "expected" behavior if the low flows are strictly environmental. However, the fact that the environmental and out-of-river diversions are grouped together means that the out-of-river diversions have priority over the out-of-reservoir diversions. If this relative priority is not desirable, the rules could be reversed and/or the out-of-river demands can be separated from low flow demands that are strictly environmental.

7. The Hydropower rule executes last. All other releases automatically go through the turbines and generate power to the maximum extent possible, so when this rule fires, it compares the hydropower already being generated at each reservoir with the specified load. If the load is not met, the rule makes an additional release to generate additional power. These releases are constrained only by power pool drawdown limits and limitations on increasing flooding downstream.

8. The Fish Release rules (where defined) execute and determine the additional outflow to meet the Fish Release demand. The rule sets the reservoir Outflow and reservoir Off Peak Spill slots. This again causes the system to re-dispatch and solve. The rule also performs accounting actions to calculate and set the inflows and outflows to the fish account on the reservoir.

Direct from reach diversions are not included in the rule set because they are designed to take whatever water is available in the reach.

3.2.4 Execution constraints to execute rule once per timestep

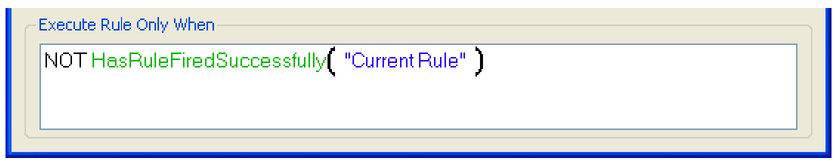


Figure 6. Execution constrain to limit firing.

In general, each rule executes only once per timestep. This is guaranteed by including an execution constraint for each rule that calls the HasRuleFiredSuccessfully predefined function. For Example, the execution constraint for the Flood Control Rule says that it will only execute when NOT(HasRuleFiredSuccessfully("Current Rule")), see Figure 6. The keyword "Current Rule" is used to reference the given rule. This prevents the user from having to change this argument if the name of the rule changes. The name of the rule, as a string, is also valid.

3.2.5 Types of rules

In rulebased simulation, rules set slots equal to the value determined by the logic on the right side of the assignment statement. For the USACE-SWD functionality, there are two variations to this: rules loop through a list of lists and set values from those two lists. Also, for Surcharge and regulation discharge, the rules set flags on slots. These two approaches are described in this section to familiarize the user with this structure.

3.2.5.1 Assigning values using Lists of Lists

The four specific USACE-SWD predefined functions (FloodControl, HydropowerRelease, ComputeReservoirDiversions, MeetLowFlowRequirements) that are accessed by the rules all return a list of lists. The outer list contains an inner list that consists of a triplet of {slot, value, and object}. For example:

{ {"Res1.Outflow", 6344.32 ["cfs"], "Res1"},

{"Res1.Flood Control Release", 6344.32 ["cfs"], "res1"},

{"Res1.Target Balance Level", 8.32, "res1"} ,

{"Res2.Outflow", 3243.02 ["cfs"], "Res2"},

{"Res2.Flood Control Release", 2312.20 ["cfs"], "Res2"}, {“Res2.Target Balance Level”, 8.32, “Res2”} }

Generally, the rule loops over this list and sets the slot at index 0 (the first one, the slot) to the value at index 1 (the second one, the value). The body of the rule would look like this:

```
FOR (LIST triplet IN FloodControl( “Flood Basin” )) DO
    ( triplet < 0 > )[] = triplet < 1 >
END FOR
```

The third item, the object) is typically only used in debugging.

3.2.5.2 Setting Flags

The second unique syntax used by the USACE-SWD is to set a flag on a slot. This applies to the Surcharge and Regulation discharge rules. In general, the slot is assigned the flag. The flags are all found on the palette. For example,

Reservoir.Outflow[ ] = SURCHARGE_RELEASE_FLAG

This assignment statement says that the Surcharge flag should be set on the Reservoir.Outflow slot at the current timestep.

3.2.6 Use of 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.

3.2.6.1 Create a Computational Subbasin

The Subbasin Manager dialog is invoked by selecting **Workspace ➡ Edit Subbasins** from the main RiverWare menu. The Subbasin Manager dialog has three tabs: **User Defined**, **Automatic**, and **Object Membership**. Computational subbasins are added in the User-Defined Subbasins tab. Then, in the Subbasin Manager dialog, select **Subbasin ➡ Append Typed Subbasin ➡ Computational**. A new subbasin will appear in the dialog box and the user can rename the subbasin if desired. The subbasin will contain any objects that were selected on the workspace. The following section describes how to add objects to the subbasin.

3.2.6.2 Adding Objects to Subbasin

All objects which will be used in the specific calculations must be included in the computational subbasin. Furthermore, all objects in the subbasin must be continuous (i.e. there can be no disconnected objects with the exception of data objects). The easiest way to add objects to the subbasin is to select the objects on the RiverWare workspace by “drawing” a large box around all the relevant objects. This can be done by starting at a corner of the model then clicking and dragging until all of the objects are highlighted. Once the objects are highlighted on the workspace, highlight the new subbasin then select **Subbasin ➡ Append Members from Workspace** in the Subbasin Manager dialog. The new subbasin should then show all of the objects associated with it when the subbasin view is expanded (by clicking on the small icon to the left of the subbasin name). Even though data objects are not part of the flood control calculations, for ease of selecting and appending objects to a subbasin their inclusion in the flood control subbasin is allowed.

3.3 Surcharge

The surcharge policy determines the mandatory release that must be made to protect the reservoir.

3.3.1 Model setup

On each reservoir, select the desired surcharge method from the Surcharge Release Category. The multi-object method selector can be used to make this selection on all reservoir objects at once.

Typically in USACE-SWD models the **Flat Top Surcharge** method is used and will be used as an example here. Once this method is made active, several slots will be added that require user input. The Rating Curves table contains the induced surcharge and free flow rating curves. The Operating Level Table contains the pool elevation vs. operating level information for each operating level. The body of this periodic slot contains the pool elevations and the column headers represent the balance levels. So there must be a column for each operating level and the pool elevations within that column may vary with time. Input a value in the Top of Flood Pool and Top of Conservation Pool slots. These values represents the operating level (not elevation) associated with the top of flood pool and top of conservation pool, respectively. They must be the same on all reservoir objects. The user can set these values on all objects at once using the computational subbasin (the computational subbasin is discussed below).

Additional information about the **Flat Top Surcharge** method can be found **HERE (Objects.pdf, Section 24.1.15.2)**. Additional information on the other **Surcharge Release** category methods can be found **HERE (Objects.pdf, Section 24.1.15)**.

If using the **Flat Top Surcharge** method, consider also selecting the **Constant Additional Surcharge Release** method in the **Elevation Max Duration Constraints** category **HERE (Objects.pdf, Section 24.1.17.2)**. This method compute the constant additional surcharge release to meet one or more maximum elevation duration constraints.

3.3.2 RPL Implementation

Since the surcharge release rules must execute first, they should be the lowest priority rules. Furthermore, the surcharge release rules must execute in an upstream to downstream order. Therefore the surcharge release rules for the most upstream reservoirs must be the lowest priority rules. Its important to remember that the lowest priority rules are the rules with the highest numbers. Conversely, rule 1 is the highest priority rule and will execute last when the agenda order is ...,3,2,1.

The surcharge release rules should be contained in their own policy group for easy organizing. Add a policy group by selecting **Ruleset ➥ Add Policy Group** in the main Ruleset Editor dialog and then name the policy group accordingly. Rules can be added to this policy group by selecting **Ruleset ➥**

Add Rule with the policy group highlighted. Add a rule for each reservoir in the basin. The rules are edited by expanding the policy group view and then double clicking on the red "R" icon next to one of the rules. This invokes the Rule Editor dialog. Each rule should be renamed accordingly. In the Rule Editor dialog, add an assignment by selecting **Rule ➡ Add Assignment**. Then, use the palette to build the rule.

Each rule should set the outflow slot of the reservoir to the surcharge release flag. This will trigger the reservoir to solve for its surcharge release forecast and set its outflows (for each timestep in the forecast period) equal to the computed surcharge releases. Since the downstream reservoirs need to know the surcharge release forecasts from the upstream reservoirs before they can solve for their surcharge releases, the rules must be ordered to execute in an upstream to downstream direction. If a reservoir receives releases from more than one upstream river branch, then each upstream branch must solve first. An example rule for one reservoir is shown in Figure 7. The end result will be a series of rules that set the outflow slot on each reservoir to the surcharge release flag. The rules will execute from lowest priority to highest priority (from the upstream reservoirs to the downstream reservoirs).

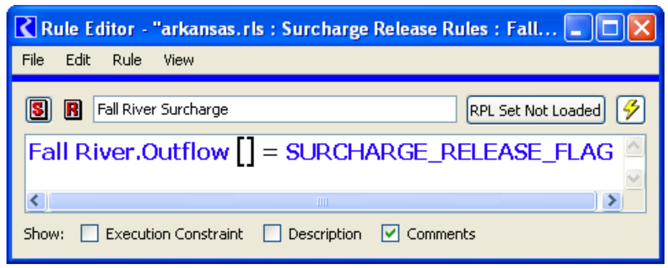


Figure 7. Sample Surcharge Rule

3.3.3 Description of logic

The logic used in the **Flat Top Surcharge** computations is presented in the object documentation **HERE (Objects.pdf, Section 24.1.15.2)**.

A description of the interaction between the rule and object dispatching is as follows: When the surcharge rule executes, it sets the surcharge release flag on the reservoir outflow slot. This triggers the object to dispatch at the current timestep (solveMB_givenInflowOutflow) and solve for its surcharge release forecast. Within this dispatch, it sets the Surcharge Release slot and also sets the surcharge release flag (and physical constraint flag) on the outflow slot at each subsequent timestep in the forecast period. This causes the object to be put on the queue at each of those timesteps. When the reservoir dispatches at each timestep in the forecast period (t), it recognizes that the controller timestep and the object timestep are not the same, and it does not re-compute the forecast of surcharge values. Instead it sets the Outflow (@t) equal to the Surcharge Release (@t) that was previously computed.

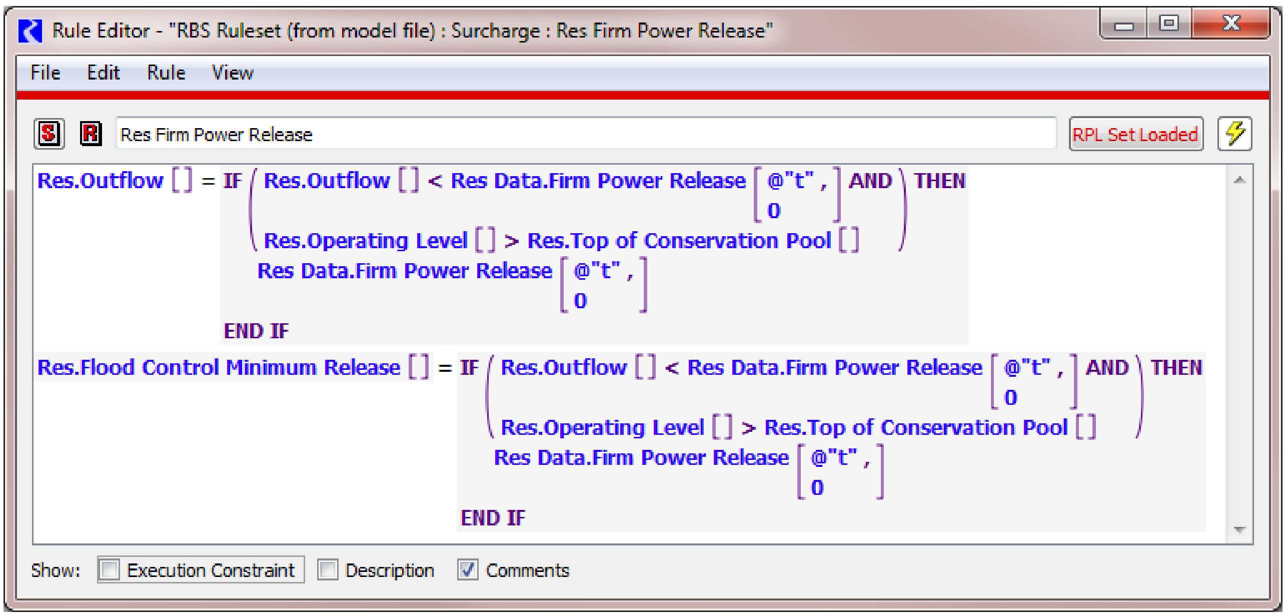
Once the outflow slot is set, the value is then routed downstream to the next reservoir. Then the next rule will execute, triggering another reservoir to solve for its surcharge releases and route downstream. This continues until all rules have executed and all reservoirs have computed their surcharge release forecasts

3.4 Flood Control Minimum Release

In certain basins, water is released from a reservoir regardless of the downstream constraints or balancing. These minimum releases are not surcharge releases but are considered minimum flood control releases. They can be modeled by setting the Flood Control Minimum Release slot (described **HERE (Objects.pdf, Section 17.1.24.2)**) and the reservoir Outflow slot with a rule. It is up to the user to write the rule logic of when these slots should be set and their corresponding values. As shown **HERE (Section 3.2.3)**, these rules should execute after surcharge, but before regulation discharge. Then, the minimum releases will be routed downstream and the Regulation Discharge rules will include them in their empty space determination.

3.4.1 RPL Implementation

It is up to you to write the rule logic to set the reservoir Outflow and Flood Control Minimum Release. Shown is a sample rule.



In this sample rule, the current Outflow and Flood Control Minimum Release is set to the Firm Power Release when the current Outflow is less than the Firm Power Release and the Operating Level is above the Top of Conservation Pool.

Note: Only the current timestep values are set by this rule. Setting future timesteps in the forecast period is possible but not easy due to the priorities of the rules.

Optionally, if you wish the Minimum Flood Control Release to not go through the release or turbines, you can also set the Regulated Spill or Bypass slot in the rule. You may need to select a regulated spill or bypass method on the reservoir to add these slots to the reservoir.

3.5 Regulation Discharge

The regulation discharge operation calculates the regulation discharge, meaning the maximum flow that is allowed in the channel at the control point, for each timestep in the specified forecast period. It also computes the empty space at the control point. Calculation of these values are fundamental to the Operating Level Balancing procedure for determining flood control releases. Thus, in effect, the regulation discharge operations are looking at the state of they system after the surcharge release to determine the available space at each control point for the flood control operation.

3.5.1 Model setup - Control Point methods

A regulation discharge method must be selected on each Control Point that will be involved in the Flood Control. There are a number of methods that are described **HERE (Objects.pdf, Section 9.1.9)**. Each method describes the slots that are instantiated and the logic that it uses to compute the Regulation Discharge and Empty Space.

> **Note:** The desired Flood Control method must be selected before the Regulation Discharge category becomes available.

Depending on the Regulation Discharge method selected, additional categories may be available. Each category has methods that modify how the Regulation Discharge is performed. The possible categories are:

- **Percent Full Determination**: **HERE (Objects.pdf, Section 9.1.10)**
- **Stage Control Over Forecast**: **HERE (Objects.pdf, Section 9.1.11)**
- **Sag Operation**: **HERE (Objects.pdf, Section 9.1.12)**
- **Regulation Recession**: **HERE (Objects.pdf, Section 9.1.13)**

3.5.2 RPL Implementation

The regulation discharge calculations are performed on the control point objects after the surcharge release forecasts have been computed and routed downstream. In order to ensure that the surcharge release calculations have finished, the rule to trigger the regulation discharge calculations must execute after all the surcharge release rules. This means it must be a higher priority rule. Add a policy group for regulation discharge in the main Ruleset Editor dialog and add a single rule to this policy group. Make sure that this policy group sits above the surcharge release policy group thereby making the regulation discharge rule a higher priority rule.

A single rule can be used to trigger the regulation discharge calculations on all control point objects. The rule should consist of a FOR loop that will loop through every control point in the flood control subbasin and set the regulation discharge flag on the **Reg Discharge Calculation** slot. When this flag is set it will trigger the control point to solve for regulation discharge and empty space. An example rule is shown in Figure 8.

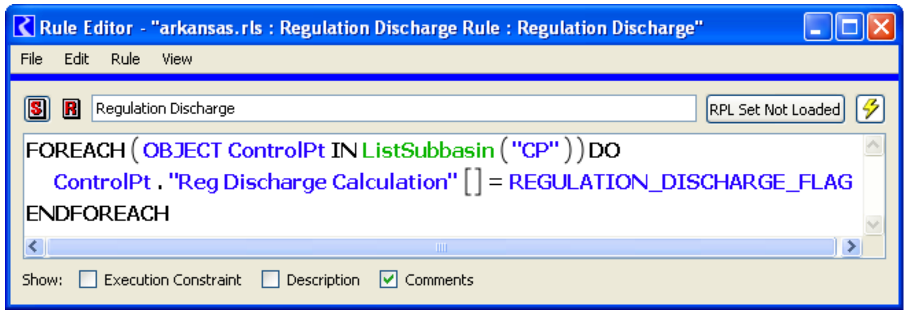


Figure 8. Sample Regulation Discharge Rule

This rule consists of a single FOR statement. The ControlPt variable is a temporary loop variable whose expression data type is OBJECT. The FOR statement will loop through every object in the list on the right hand side of the IN expression and the ControlPt variable will become each object in that list. The right hand side of the IN statement is a predefined function named ListSubbasin that takes a subbasin as an argument and returns a LIST that contains each object in the subbasin. In this example, the subbasin used in the ListSubbasin function is called “CP” and is a user defined subbasin. The rule could have used the “ControlPoint” automatic subbasin. This subbasin can be viewed in the Automatic Subbasins view of the Subbasin Manager dialog. If every control point in your model is used in flood control, then your rule can look just like the example rule. If however the user is only using a subset of control points for the flood control calculations, a new subbasin must be created that contains the subset of control points. The rule should then refer to the new subbasin in the ListSubbasin predefined function. In summary, the example rule will loop through every control point in the specified subbasin and set the Reg Discharge Calculation slot on each object (at the current timestep) with the regulation discharge flag. This will trigger each control point to perform its regulation discharge calculations and compute the empty space for the forecast period. Then the model is ready to compute the flood control releases.

3.6 Flood Control

The Flood Control methods calculate a flood control release for the current simulation day. The flood control methods 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 to account for routing effects. They return the values for the first timestep of the proposed schedule (for the current simulation timestep).

3.6.1 Model setup

There are two flood control methods, **Operating Level Balancing** and **Phase Balancing**. **Phase Balancing** is described in detail on the computational subbasin, **HERE (Objects.pdf, Section 7.1.3.3)**. Because most of the Southwest Division uses the **Operating Level Balancing** method, it is the only one discussed in this document.

To use the **Operating Level Balancing** method, the user must select methods on each reservoir and control point. Also a computational subbasin must be created that contains the desired objects (typically all of the objects in the model). The user must also select a number of methods on this computational subbasin. Below is an overview of the methods that should be considered on each object type:

3.6.1.1 Reservoir methods

On each reservoir, select:

- **Operating Level Balancing** method in the **Flood Control Release** category, **HERE (Objects.pdf, Section 24.1.18.2)**
- **Conservation and Flood Pools** in the **Operating Levels** category, **HERE (Objects.pdf, Section 22.1.19.3)**

The multi-object method selector can be used to make this selection on all reservoir objects at once. The flood control method does not perform calculations on the object, but provides several slots that will be added that require user input:

- Allowable Rising Release Change
- Allowable Falling Release Change
- Maximum Release Variation
- Balance Period
- Forecast Period
- Top of Flood Pool
- Top of Conservation Pool
- Bottom of Conservation Pool
- Operating Level Table

The value entered in the Balance Period and Forecast Period slot must be the same on all reservoir objects used in flood control. The user can set these values on all objects at once through the computational subbasin (**HERE (Section 3.6.1.3)**).

Optionally, you can select a method in the Conditional Operating Level category (**HERE (Objects.pdf, Section 22.1.20)** to add an alternative operating level table, which will be used when hydrologic conditions indicate.

3.6.1.2 Control Point methods

On each control point, select the **Operating Level Balancing** method in the **Flood Control** category, **HERE (Objects.pdf, Section 9.1.8.2)**. The multi-object method selector can be used to make this selection on all reservoir objects at once.

Once **Operating Level Balancing** method has been selected, several other method categories become active. These include, but are not limited to:

- **Key Control Point Balancing**: **HERE (Objects.pdf, Section 9.1.14)**

- **Flooding Exception**: HERE (Objects.pdf, Section 9.1.15)

The user must decide if any of these categories apply to the control point in question (a selection is not necessary if these calculation do not apply). Several slots may be added as a result of these method selections.

If the control point is a key control point, select a method in the Key Control Point Balancing category. If a method is selected, the control point will be displayed as a key control point on the workspace with a "key" icon in the center of the object. Control point objects designated as key control points will be included in the reservoir balancing calculations performed during the computation of flood control releases.

3.6.1.2.1 Required Input Data

Configure the Upstream Reservoirs and Routing Coefficients slots on each control point. The Upstream Reservoirs slot is a list of all upstream reservoirs whose releases are routed through the control point in question. The Routing Coefficients slot will have a column for each reservoir listed in the Upstream Reservoirs slot. In each column, enter the direct routing coefficients from the reservoir in the column header to the control point object. These coefficients will be used to route releases in the algorithm that computes the flood control releases. The actual routing of the final releases is done incrementally via the reach objects. Therefore, in order to have consistent routing between the flood control algorithm and the reach objects, these direct routing coefficients should be computed from the incremental routing coefficients used on the reach objects (or vise versa) using a sufficient number of significant figures.

> **Note:** Flow based routing coefficients are allowed. Click HERE (Section 2.10) for more information on these methods.

3.6.1.2.2 Key Control Point Balancing

If the control point is a key control point, select the Operating Level Balancing method in the Key Control Point Balancing category (the Control Point icon will then display a key symbol). Several slots are added when this method is selected. Enter a list, in the Key Control Point Reservoirs slot, of the upstream reservoirs that are balanced by the control point. This list should either be the same as, or a subset of, the list of reservoirs in the Upstream Reservoirs slot. Input a value for Balance Period. This value must be the same value on all key control points and reservoir objects. The computational subbasin can be used to set the value on all objects simultaneously. Also, input values for the Balance Tolerance and Balance Iterations slots. The affects that these slots have on the balance calculations are discussed HERE (Objects.pdf, Section 9.1.14.1).

3.6.1.3 Computational Subbasin methods

Create a Computational Subbasin for flood control using the Subbasin Manager dialog and specify the objects included in the subbasin. The subbasin consists of all the objects, linked to form a continuous network, that are considered in the simultaneous flood control release calculations.

3.6.1.3.1 Configure Subbasin for Flood Control

Once the computational object has been created it can be configured for flood control. The computational subbasin object can be opened in two ways: by double clicking the subbasin in the Subbasin Manager dialog, or by selecting **Workspace ➥ Open Computational Subbasin** from the main RiverWare menu.

The open subbasin dialog has two tabs: Slots and Methods. The slots view will be empty until a method has been selected. In the Methods view, select the **Operating Level Balancing** method in the **Flood Control** category. Several new slots and method categories will be added as a result of this selection. Input values in the following slots:

- Forecast Period
- Balance Period
- Top of Conservation Pool
- Highest Operating Level
- Lowest Operating Level
- Top of Flood Pool.

The remaining slots with the word “Tolerance” in the name will default to the tolerance values used in SUPER. Details on these slots can be found **HERE (Objects.pdf, Section 7.1.3.3)**.

Also, when the **Operating Level Balancing** method is selected, a number of categories become available. All of these dependent method categories on the computational subbasin default to the method selection that corresponds to SUPER behavior. Details on these methods can be found starting **HERE (Objects.pdf, Section 7.1.4)**.

3.6.1.3.2 Propagate Values

The values entered in the Forecast Period, Balance Period, Top of Flood Pool and Top of Conservation Pool slots must be the same on all objects that have these slots. The user can enter these values on the computational subbasin and then propagate the values to all other objects using the **Subbasin ➥ Propagate Values** feature in the open subbasin dialog.

3.6.1.3.3 Verifying and Enabling Subbasin

Once the subbasin has been configured, it can be verified and enabled. Subbasin verification is performed by selecting **Subbasin ➥ Verify Subbasin** in the open subbasin object. This is an internal check to ensure that the subbasin is continuous and that all the required data has been input. Once the subbasin is ready for use in a model run, it can be enabled by selecting **Subbasin ➥ Enable Subbasin** in the open subbasin object. When a subbasin is enabled, the verification checks are performed at the beginning of the run and the subbasin can be used by the Flood Control function. If the subbasin is not being used in a model run and will not pass verification, it can simply be disabled and will not hinder the model run.

3.6.2 RPL Implementation

Once the surcharge release forecasts have been computed and routed downstream, and the empty space forecasts have been computed, the model is ready to solve for the flood control releases. The flood control logic is called from the FloodControl predefined function. This function will perform the flood control calculations for all objects in the specified computational subbasin. After execution, the FloodControl function will return the flood control value and the outflow value, for the current timestep, for each reservoir in the computational subbasin. The flood control rule will set these values on each reservoir object. The resulting reservoir outflow is then routed downstream.

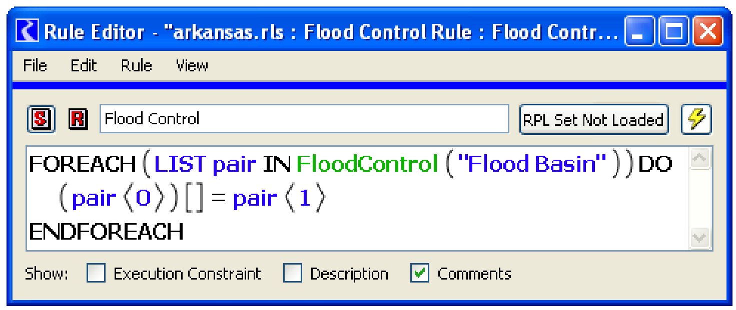


Figure 9. Sample Flood Control Rule

The flood control rule must execute after the regulation discharge calculations and therefore must be higher priority. Add a new policy group above the regulation discharge policy group and add a single rule to this policy group. Figure 9 shows an example rule that executes the FloodControl predefined function and sets the resulting outflow and flood control release (and possibly Target Balance Level) on each reservoir in the Flood Basin subbasin.

The FloodControl predefined function executes the flood control algorithm. While it computes the flood control releases over the forecast period (using the inflow, surcharge release, flood control minimum release and empty space forecasts computed previously), only the final flood control release for the current timestep is set in the model. Therefore, the FloodControl function returns the flood control release and outflow values for the current timestep only. The FloodControl function returns a LIST that contains three items for each reservoir. Each list contains a slot, the value that should be set on that slot, and the object. So for each reservoir, there are three slots that are typically set using a FOR statement:

- Outflow = Surcharge Release + Flood Control Release + Flood Control Minimum Release
- Flood Control Release
- Target Balance Level

The example rule above consists of a FOR statement that calls the FloodControl function and sets the resulting values on the proper reservoir slots. The FloodControl function performs the flood control calculations for all objects in the Flood Basin subbasin over the forecast period. In this example, Flood Basin is the name of the computational subbasin created for flood control (this may be named differently in your model). The FloodControl function returns the lists described above and the FOR expression loops through each list and sets the returned slot, at the current timestep, to the corresponding value.

In summary, the flood control rule calls the FloodControl function which computes the flood control releases for every reservoir in the computational subbasin. Then, for each reservoir, at the current timestep, it sets the Flood Control Release slot to the computed value, the Outflow slot to the computed value, and the Target Balance Level to the computed value. After the rule is finished executing, the new outflow values are routed downstream and all objects in the model re-solve given the outflow values.

For more information on the function itself, click**HERE (RPLPredefinedFunctions.pdf, Section 40)**.

3.6.3 FloodControl() function - Detailed description of logic

The logic for this function is presented in the computational subbasin help **HERE (Objects.pdf, Section 7.1.3.3)**.

3.7 Low Flow Releases

The Low Flow Release policy computes the reservoir outflows to meet a low flow demand at a downstream control point. If a control point's demand can be met by multiple reservoirs, the fullest reservoir is considered first and releases as much as possible, then the next reservoir is considered. To set up this operation, the user must select methods and implement rules that call the predefined MeetLowFlowRequirements function.

3.7.1 Model setup

The user must select methods on each control point and reservoir involved in the low flow operation. Also, a computational subbasin must be created and methods must be selected on that subbasin.

3.7.1.1 Control Point methods

For a control point with a low flow requirement, the user should select a method in the **Low Flow Requirement** category to specify how the demand is represented. The valid options are:

- **Low Flow Periodic Lookup**: The demand is specified in a periodic slot **HERE (Objects.pdf, Section 9.1.17.2)**
- **Reservoir Level Lookup**: The demand is specified in a periodic slot that relates a reservoir level to the demand. **HERE (Objects.pdf, Section 9.1.17.3)**. If the specified reservoir is disabled and is set to Pass Inflows, no low flow requirement can be computed. See the note **HERE (Section 4.3.4)**.

Both of these methods add the slot Low Flow Reservoirs which is used to specify the upstream reservoirs that can serve the low flow demand.

3.7.1.2 Reservoir methods

On each reservoir that can possibly serve a Low Flow demand on a control point, the user should select the **Enable Low Flow Releases** method in the **Low Flow Releases** category, **HERE (Objects.pdf, Section 24.1.14.2)**. This method adds the Low Flow Release slot to hold the result of the policy and the Maximum Low Flow Delivery Rate which is the maximum flow that can be released from the reservoir to meet low flow demands.

3.7.1.3 Computational Subbasin methods

A computational subbasin should be created that contains the objects involved in the low flow release. Typically, the same subbasin used for the Flood Control rule can be used. On this subbasin, the user should select the **Operating Level-Based** method in the **Low Flow Releases** category **HERE (Objects.pdf, Section 7.1.2.2)**.

3.7.2 RPL Implementation

A rule will need to be created for each control point that has a low flow requirement. In the rule, a predefined RPL function, MeetLowFlowRequirement **HERE (RPLPredefinedFunctions.pdf, Section 120)**, is used to compute the required low flow releases for each reservoir. The function will take a computational subbasin and a control point as an argument.

Following is a screenshot, Figure 10, showing an example of the rule that meets the Eagletown low flow requirement using the computational subbasin name FloodBasin. The execution constraint forces the rule to execute only once.

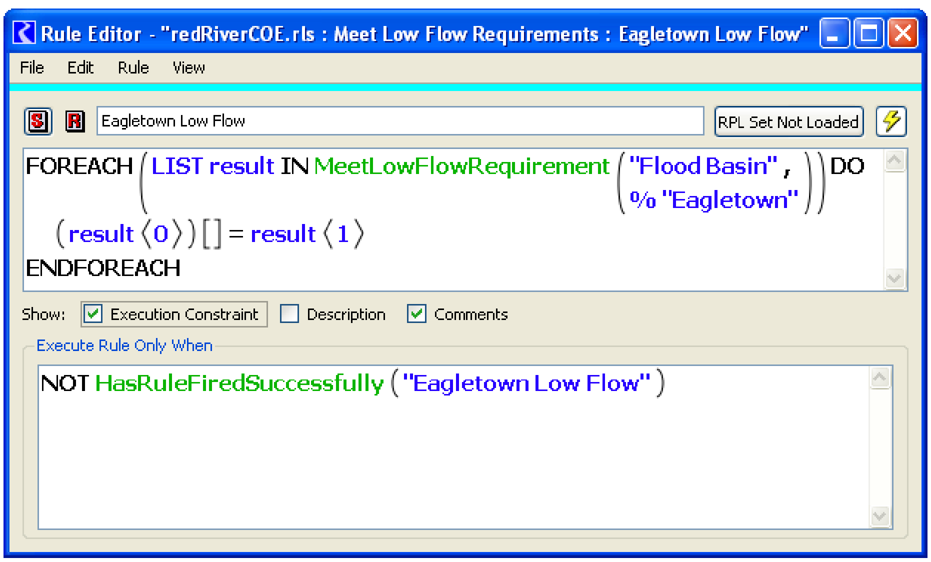


Figure 10. Sample Low Flow Release Rule

3.7.3 MeetLowFlowRequirements function

The details of the predefined function MeetLowFlowRequirements is described **HERE (RPLPredefined-Functions.pdf, Section 120)**.

The function iterates through each of the Low Flow Reservoirs specified on the control point. For each control point (i.e. each time the RPL function is called), the following steps will take place:

- The Low Flow Deficiency slot is checked to see if there is a low flow shortage. If not, the Low Flow Release is zero for each reservoir. The Outflow Should not be reset if the Low Flow Release is zero.
- If there is a deficiency, the following steps are taken:

1. The reservoirs in the Low Flow Reservoirs slot are sorted by level in descending order. Reservoirs below the bottom of the conservation pool are excluded.

2. Each reservoir (beginning with the most full reservoir) makes a release until the requirement (in the Computed Low Flow Requirement slot) is met, the Maximum Low Flow Delivery Rate (on the reservoir) is met, or the reservoir reaches the bottom of the conservation pool (whichever value is lowest). Routing coefficients (travel times) are not considered.

3. The RPL function returns a list of paired items, the Low Flow Release slot and the value to be set on the slot (for each reservoir). The rule makes the assignment to the Low Flow Release slot. The new Outflow and Outflow slot is also returned by the rule function. A ForEach statement is used to make the assignments.

When the reservoir Outflows are set, the system will re-solve to route the flows downstream. The Low Flow Deficiencies will be recomputed to reflect the low flow releases. The reservoir Operating Levels will be recomputed to reflect the new releases. Then the next rule will execute to determine the low flow releases for the next control point.

3.7.4 Setup to avoid routing low flow releases

In the MeetLowFlowRequirement function, routing is not considered, but in the subsequent simulation of the releases, routing is considered. As a result, the released flow often does not meet the demand. This section describes the issues with the approach, then presents a structure to avoid routing of the low flow releases.

Releases are made from reservoirs to meet minimum flow criteria specified at downstream Control Points. In SUPER, routing is considered in the low flow release calculation as follows: *Today's low flow release fills deficiencies noted in yesterday's final results. Yesterday's final results are adjusted by today's low flow release.* In RiverWare’s MeetLowFlowRequirement function, the current timestep’s low flow release deficiency is used as the low flow demand and water is released from the reservoir to meet this demand, but routing is ignored. The following example illustrates the issue:

Assume a control point that is 1 day travel time downstream of the reservoir with a 1000 cfs low flow requirement every day. Assume there are no other releases, no surcharge or flood control, and hydropower is inactive. On the current day, let's say there is 500 cfs at the control point from uncontrolled area flows. The reservoir would make a 500 cfs low flow release to make up for that deficiency. However, that release would not get there until the next day. On the next day, there are no other releases or inflows. So the low flow deficiency is still 500 cfs (the requirement is 1000 cfs and there are 500 cfs at the control point from the previous day). The reservoir would again make a 500 cfs release to cover the deficiency. However, that wouldn't get there until the next day. On the next day there is still a 500 cfs deficiency (let's keep assuming there are no other releases and no uncontrolled area inflows). So the reservoir would again make a 500 cfs release.

This example illustrates the problem with the existing RiverWare approach - the minimum flows may never be satisfied. RiverWare tries to make up the current timestep's deficit, which often does not arrive at the control point until future timesteps. This leads to less water being released. In general, the releases for this purpose are smaller in RiverWare than in SUPER.

The model can be modified to address low flow issues in a manner consistent with SUPER low flow handling. That is, reservoir releases made specifically for low flow requirements at a downstream control point are passed through secondary flow paths avoiding routing between the reservoir and control point. Below is the approach.

3.7.4.1 Creation of Secondary Flow Paths

To avoid routing the effects of low flow releases, designated low flows were passed around routing reaches by creating a secondary flow path between a control point and its associated upstream reservoir(s) as follows:

1. a non-routing reach was created for use in diverting the designated low flows from the primary (existing) flow path.

2. a water user was created to the side of the primary flow path, facilitating diversion of water from the non-routing reach and returning flows to the routing reach's Return Flow slot.

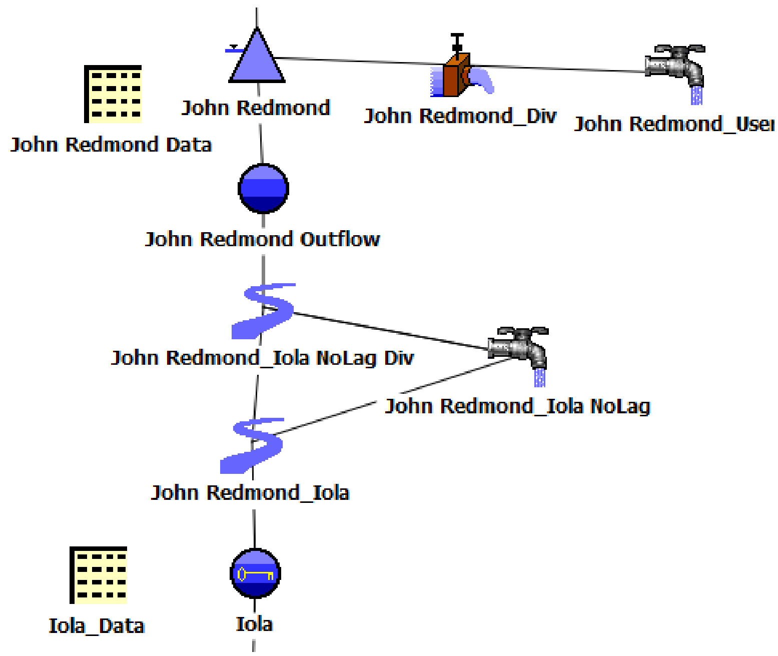


Figure 11. Sample setup to avoid routing low flow releases

Figure 11 shows this setup. The secondary flow path was created by linking these two objects. The non-routing reach Available For Diversion slot was linked to the water user Incoming Available Water slot. The non-routing reach Diversion slot was linked to the water user Diversion slot. The water user Return Flow slot was then linked to the routing reach's Return Flow slot which flows into the routing reach after the main channel flow has been routed.

The non-routing reach was then inserted into the existing flow path, with existing links being restructured to provide inflow / outflow connectivity. The outflow / inflow link connecting the routing reach object and its upstream object(s) was deleted. Outflow from the upstream object(s) instead was linked to the inflow of the new non-routing reach. Outflow from the new non-routing reach was linked to the inflow of the routing reach.

In the case where a reservoir provides low flow releases to two (or more) downstream control points, this structure must be created around each routing reach between the reservoir and the lowest control point. Flows destined for low flow demand at the lowest control point should pass through secondary flow paths created for flows destined for control points higher up in the river system. It is not necessary to create more than one secondary flow path around a single routing reach, the rules will determine the quantity of water to divert through the secondary path. For this setup (in the initial implementation), each reservoir should have a companion data object. On the data object, you should add slots to hold the low flow releases for each downstream control point: Reservoir Data.LowFlow for CP. For example, there may be a slot "John Redmond Data.Low Flow Release For Iola"

3.7.4.2 Method Selection on Objects

On each of the new water user objects, the Return Flow category was set to Fraction Return Flow, and then the Fraction Return Flow Input category was set to Periodic Fraction. A value of 1.0 was entered into the Periodic Fraction periodic table. This sets up zero consumption in the water user; all diversion becomes return flow.

On each new non-routing reach object (named with *NoLag Div in the sample model), the Routing method was set to No Routing. Also, on each of the new water users (named with *NoLag in the sample model) the Periodic Diversion Request method was selected from the Diversion and Depletion Request category. A value of 0.0 was input into the only cell in the Periodic Diversion Request slot. This method executes at the beginning of the run and on each timestep looks up a value in the Periodic Diversion Request slot. The computed value is set in the Diversion Requested and Depletion Requested slots. This serves to initialize the water users so their diversion request is zero at the beginning of the run.

Local Inflow and Solution Direction was set to No Local Inflow, Solve Outflow on each of the new non-routing reaches. This was discovered as necessary due to rule execution and convergence errors during simulation resulting from an unintended solution direction caused by the lagging reaches.

3.7.4.3 Rules to Divert to "no lag" Secondary Flow Paths

When using this setup, a slightly different structure should be used for performance and readability. For each control point where the flows should not be routed, there should be two rules:

1. A lower priority rule computes the necessary low flow release for the specified control point, but sets the value on the Reservoir's data object slot. No dispatching happens after this rule. An example is shown in the following screenshot

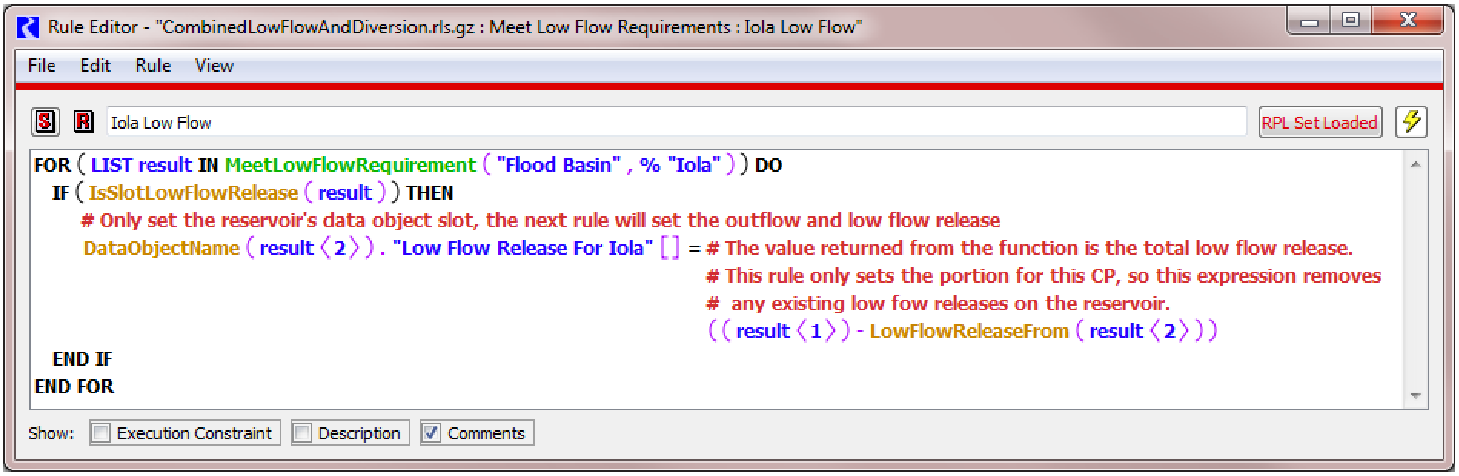
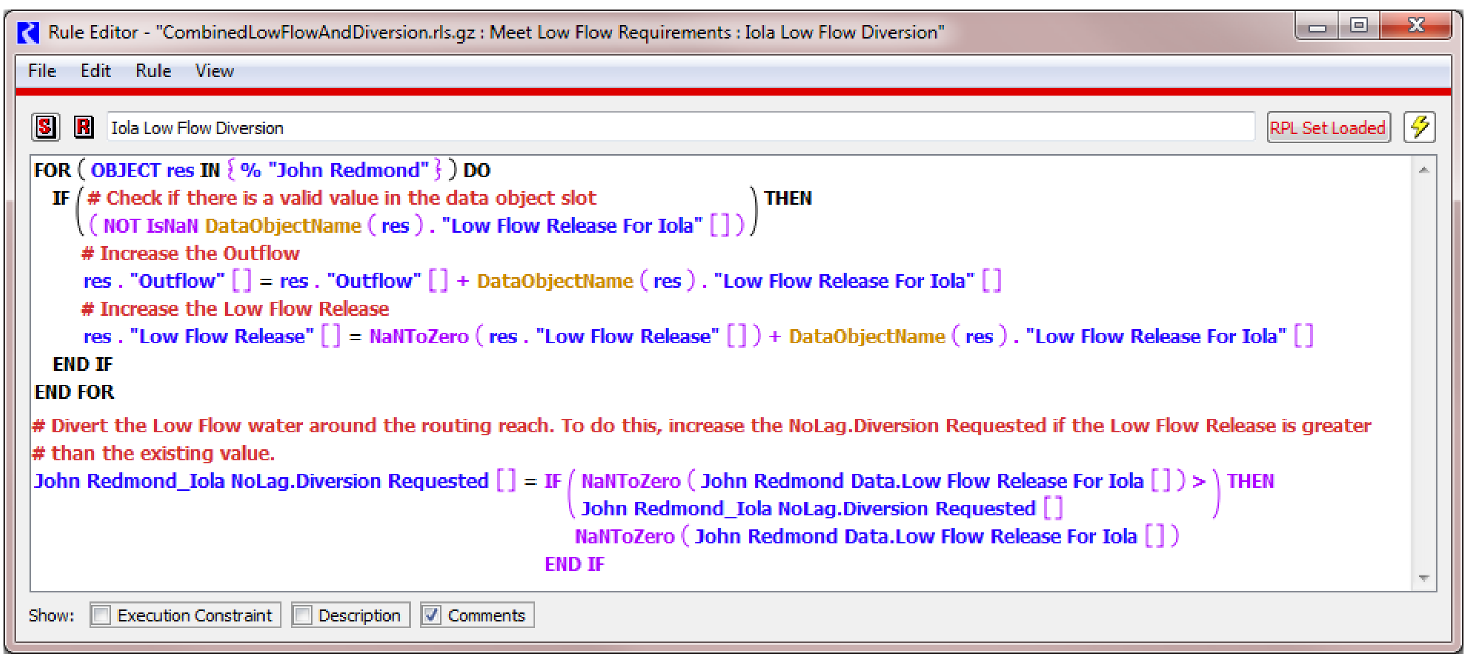


Figure 12. Sample Rule to compute low flows needed and set it on a data objects

2. The higher priority rule that applies it to the reservoir by increasing the Outflow and Low Flow Release slot. It also increases the NoLag.Diversion Requested slot. The system will dispatch and solve after this rule finishes. An example is shown below.

Figure 1. Sample Rule t o set the low flows on the reservoir and divert them around the routing reach.



This two rule structure is necessary for performance and readability. Computing the values in the first rule but setting them on the reservoir in the next rule improves performance by only dispatching the system once. These rules should be set to execute only once through execution constraints. Care must be taken if there are multiple low flow demands. The low flow diversion rule must divert the correct amount.

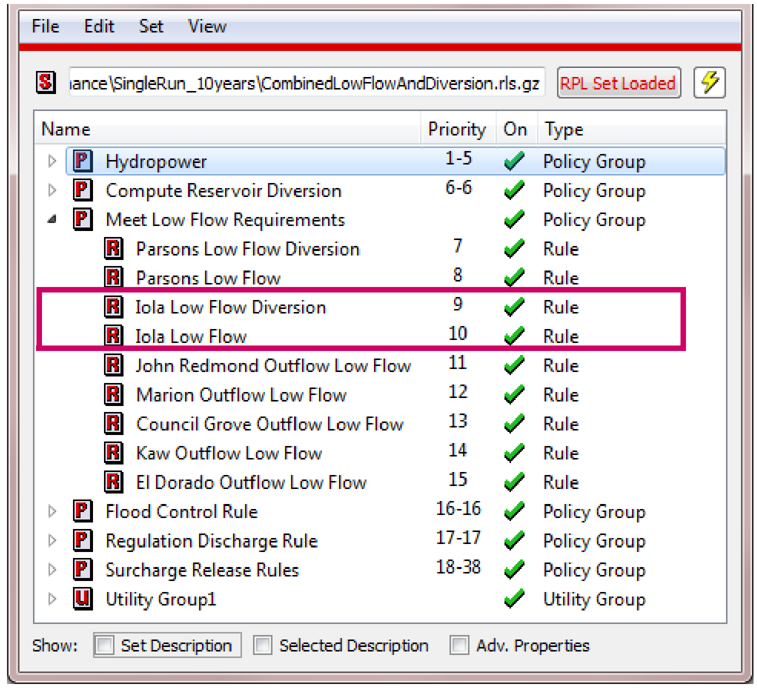


Figure 13. Sample ruleset that avoids routing low flow releases

3.8 Reservoir Diversions

Diversions are made directly out of a reservoir to meet a demand. Demands can be a water user or another reservoir. Each demand can draw from multiple reservoirs and each reservoir can act as a source for multiple water users or demand points.

3.8.1 Model setup

A diversion consists of a Reservoir, Diversion object, and Water User. In the simplest case, there is one of each as shown in Figure 14. A specific linking structure must be employed between the reservoir, diversion object, and water user objects. This can only be set up after the appropriate methods have been selected. Thus, the method selections are described first, then the linking structure is re-visited. Each of these objects must have methods selected as described in the following sections.

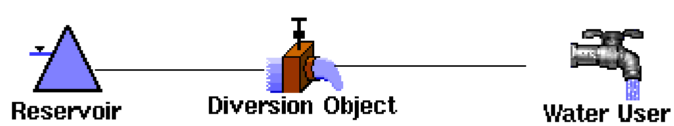


Figure 14. Sample Diversion from Reservoir setup

Note: If the reservoir is disabled and is set to Pass Inflows, no diversions are made. See the note **HERE (Section 4.3.1)**.

3.8.1.1 Reservoir methods

There are no required method selections on the reservoir objects as the Diversion slot exists on all reservoirs. In previous versions, it was necessary to select the **Available Flow Based Diversion** from the **Diversion from Reservoir** category. This is no longer necessary nor encouraged as the RPL function ComputeReservoirDiversions calculates the available flow, which is different than the Available for Diversion slot used in this method.

3.8.1.2 Diversion Object methods

On the Diversion object, the **Solve Given Outflow** should be selected from the **Diversion Object Solution Direction** category. This tells the object that the outflow will be set and cause the object to dispatch. This method is described **HERE (Objects.pdf, Section 12.1.1.2)**.

Also, the **Available Flow Diversion** method should be selected from the **Available Flow** category. This category adds the Max Diversion and Min Diversion slots which must be user input.

3.8.1.3 Water User methods

On the water user, the user must select a method from the following categories:

Category	Method	Link
Return Flow	**Fractional Return Flow**	**HERE (Objects.pdf, Section 27.1.4.2)**
Fraction Return Flow Input	**Input Fraction**, **Zero Fraction**, or **Periodic Fraction**	**HERE (Objects.pdf, Section 27.1.5)**
Diversion and Depletion Request	use either **Input Request**, **Periodic Diversion Requests**, or **Reservoir Level Lookup**.The last one allows the Diversion Request to be based on the operating level of a reservoir.	**HERE (Objects.pdf, Section 27.1.1)**
Multiple Supply Sources	**Multiple Supply Reservoirs:** Even if there is only one source, this adds the Supply From Reservoirs slot which will be used in the linking step below. It also adds the Maximum Delivery Rates which specifies the maximum flow that this water user can take.	**HERE (Objects.pdf, Section 27.1.10.2)**

3.8.1.4 Computational Subbasin methods

A computation subbasin must be defined that contains all of the reservoir, diversion object and water users that are part of the system. Typically, this can be the same subbasin used for flood control. On this subbasin, the user should select the **Operating Level-Based** method from the **Diversions from Reservoirs** category, **HERE (Objects.pdf, Section 7.1.1.2)**.

3.8.1.5 Linking Setup

Figure 15 shows the linking setup that must be used. The Diversion slot on each reservoir is linked to the Diversion slot on the Diversion Object. Do NOT link Reservoir.Available For Diversion to DiversionObject.Available For Diversion as this unnecessary and can lead to excessive dispatching.

The demands are represented by the Water User objects. Thus, 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.

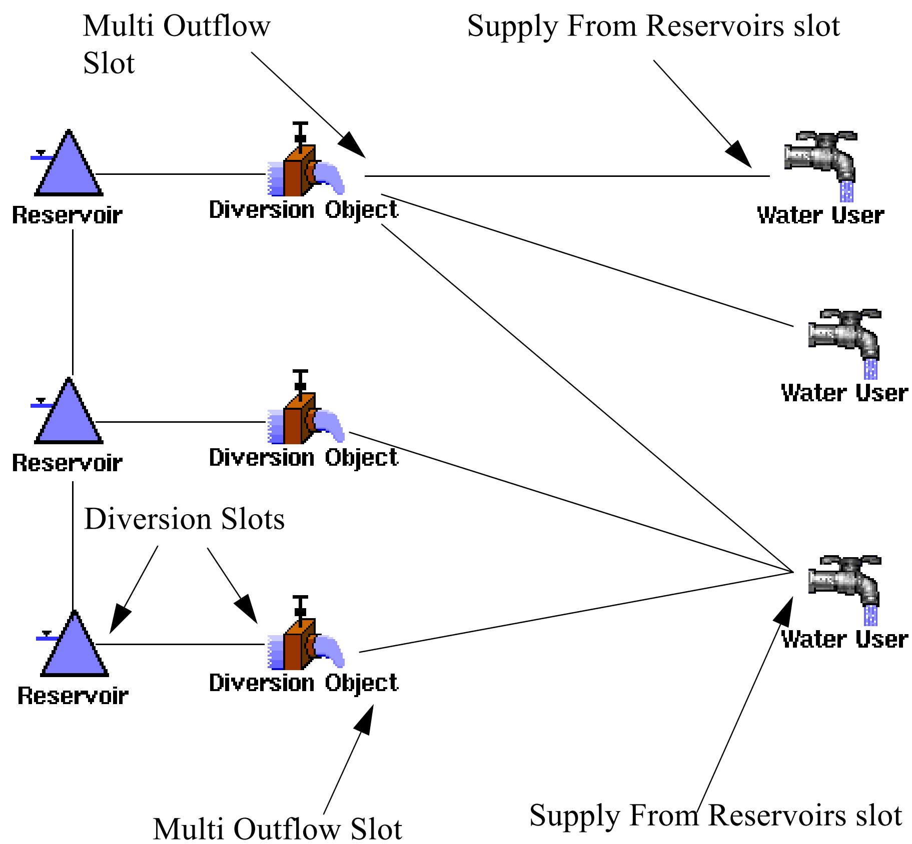


Figure 15. Schematic Diagram for ComputeReservoirDiversions Function:

3.8.2 RPL Implementation

A rule will need to be created for each subbasin that has reservoirs that divert. Typically, the same basin used for flood control can be used. In the rule, a predefined RPL function, ComputeReservoirDiversions **HERE (RPLPredefinedFunctions.pdf, Section 20)**, is used to compute the diversion from each reservoir and the supply to each water user. The function takes a computational subbasin as an argument.

Following is a screen-shot showing an example of the rule that meets the Reservoir Diversion demands using the computational subbasin name FloodBasin. The execution constraint forces the rule to execute only once.

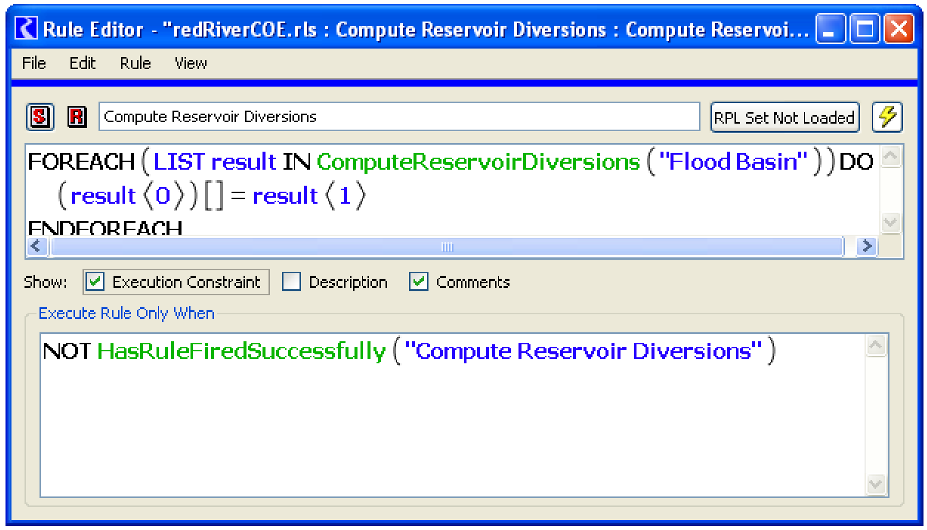


Figure 16. Sample Reservoir Diversion Rule

3.8.3 ComputeReservoirDiversions() function - Detailed description of logic

When the rule executes the ComputeReservoirDiversions predefined function, **HERE (RPLPredefined-Functions.pdf, Section 20)**, the following occurs:

Each Water User in the subbasin is visited (the order does not matter since these objects are diversions and have no upstream/downstream orientation). For each Water User the following steps are taken:

- A list of supplying reservoirs is generated based on the links to the Supply From Reservoirs slot. The links are followed to the diversion object then to the reservoir.
- The reservoirs are ranked by Operating Level (highest level first). Reservoirs below the bottom of the conservation pool are ignored.
- Loop through each reservoir. If the Limit by Reservoir Level method is selected (on the Water User object), and the Demand Reservoir has a higher level than the supply reservoir, move on to the next Water User object. If the Demand Reservoir is in the flood pool, skip ahead to the next Water User object.
- The appropriate subslot (corresponding to the link to the current reservoir in the list) is computed as the minimum of the
    - Diversion Requested,
    - Maximum Delivery Rate corresponding to that subslot (in the Maximum Delivery Rates table slot on the Water User), or
    - The amount of water that would draw the reservoir down to the bottom of the conservation pool
- Each reservoir is visited until the Diversion Requested is met or there are no more reservoirs.
- Move on to the next Water User object.

The reservoir levels are updated when moving to each water user object because multiple Water Users can divert from the same reservoir. Also, Remember, this computation is all done within the predefined function, no values on the workspace have been set.

The RPL function returns a list of slot, value pairs - the subslot and the value to be set on the subslot and also the Incoming Available Water slot and the total of all the subslots.

The rule sets the subslots on the Supply From Reservoirs slot and the Incoming Available Water slot. The Water User object can then dispatch solveStandAlone_givenDivDep (given Diversion Requested and Incoming Available Water).

The Supply From Reservoir slot propagates to the Multi Outflow slots on the connected Diversion Objects. The Diversion objects solve for their Diversion slot solveMB_givenOutflow. The Diversion values are passed to the Diversion slot on the Reservoir object and the water is removed from the Reservoir.

3.9 Hydropower

A power reservoir may release additional water from its power pool to meet a specified power commitment, i.e., the load specified for that reservoir.

3.9.1 Model setup

The user must select methods on each power reservoir involved in the hydropower operation. Also, a flood control computational subbasin must be used. Optionally, methods can be selected on the control point to limit the effect the control point has on the hydropower operation.

3.9.1.1 Power Reservoir methods

On each Level Power Reservoir, the user should select the following methods:

Category	Method	Link
Power	Peak Power Equation with Off Peak Spill	**HERE (Objects.pdf, Section 17.1.1.10)**
Power Release	Peak Power Equation with Off Peak Spill Release	**HERE (Objects.pdf, Section 17.1.2.5)**
Tailwater	any method except None	**HERE (Objects.pdf, Section 17.1.7)**
Additional Hydropower Release	Meet Hydropower Load	**HERE (Objects.pdf, Section 17.1.36.2)**
Load Calculation	any method except No Method	**HERE (Objects.pdf, Section 17.1.37)**

Each of these methods has its own data requirements as described in the appropriate section. Special consideration should be given to the Load Calculation as there are methods to specify the Load as an input, monthly value, annual value, periodic table, or seasonal weekday or weekend load or load time.

3.9.1.2 Control Point methods

On control points, consider selecting the **Flooding does not constrain hydropower release** in the **Hydropower Flooding Exception** category. This method, sometimes used on outflow control points, indicates that additional flooding at the control point does not limit hydropower releases. Note, this is not typically used except in special circumstances.

3.9.1.3 Computational Subbasin methods

Because the hydropower release operation prevents releasing hydropower water that causes additional flooding downstream, it relies on verification and data that is implemented during the Operating Level Balancing flood control initialization. As a result, the basin must execute Operating Level Balancing flood control and the same subbasin must be used for both operations. Because the hydropower operation is dependent on having flood control, no additional method selections are required on the subbasin.

3.9.2 RPL Implementation

A rule will need to be created for each flood basin that has hydropower reservoirs that make additional hydropower releases. The same basin used for flood control must be used. In the rule, a predefined RPL function, HydropowerRelease **HERE (RPLPredefinedFunctions.pdf, Section 96)**, is used to compute the additional release from each reservoir.

Following is a screenshot showing an example of the rule that meets the HydropowerRelease using the “Flood Basin” computational subbasin. The execution constraint forces the rule to execute only once.

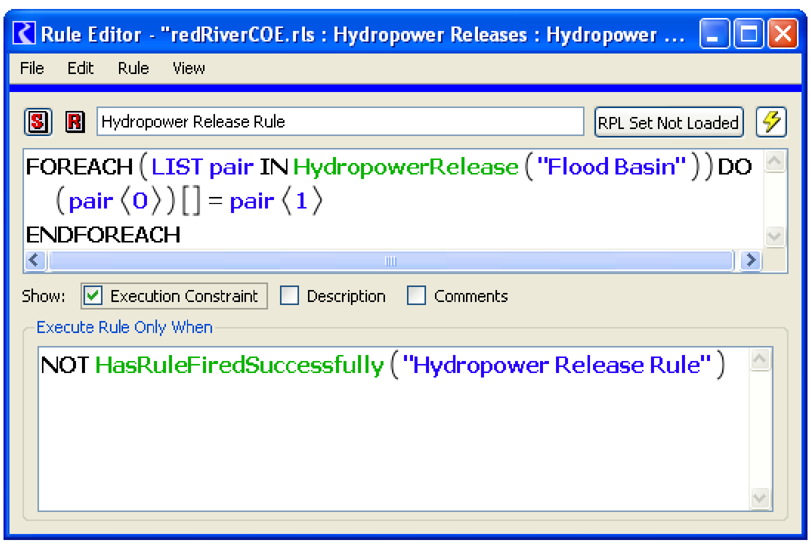


Figure 17. Sample Hydropower Release Rule

3.9.3 HydropowerRelease() function - Detailed description of logic

The general logic for the HydropowerRelease function is presented **HERE (RPLPredefinedFunctions.pdf, Section 96)**. A more detailed description of the HydropowerRelease() function is as follows.

Note: If the level power reservoir is disabled and just passing inflows as described **HERE (Objects.pdf, Section 24.1.30.2)**, no load is calculated nor are any releases made for hydropower purposes. Essentially, the disabled reservoir is skipped in all calculations described below. Also, see the note **HERE (Section 4.3.3)** about disabled downstream reservoirs limiting upstream hydropower releases.

The function:

1. Prioritizes the power reservoirs in the basin according to the relative Load shortage. This is calculated using Equation 4. This is a value less than one. The reservoirs with the highest values are first, the lowest reservoirs last.

$$\frac{\text{Load} - \text{Energy}}{\text{Load}} \qquad \textbf{(EQ 4)}$$

2. In order of priority, hypothetically calculates the additional release to meet the shortage as the minimum of the following:

- The Outflow calculated to meet the given **Load**. The method uses the user selected methods on the reservoir, i.e. it calls the Peak Power Equation with Off Peak Spill Release method and the selected tailwater method.
- The Outflow calculated such that the Pool Elevation would exactly equal the **Minimum Power Pool Elevation**.
- The Outflow calculated such that the Pool Elevation would exactly equal the **Minimum Elevation for Power Operations**.
- The Outflow calculated such that the Pool Elevation exactly equals the previous Pool Elevation minus the **Maximum Power Pool Drawdown** slot.
- 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.

3. The function then hypothetically makes the release and visits downstream control points until it reaches a tandem reservoir or the end of the subbasin, whichever comes first[1]. If the release causes (additional) flooding at a control point, it reduces the release until flooding is not caused or the release becomes zero.

1.Hydropower releases from upstream reservoirs will accumulate in the first downstream tandem reservoir, if one exists. No consideration is given to control points downstream of a tandem reservoir. The search continues through a disabled reservoirs as described **HERE (Section 4.3.3)**.

4. A control point's available space hydrograph (in units of flow projected into the future based on the routing coefficients on the control point) is calculated as:

$$\text{Available Space} = \text{Regulation Discharge} - (\text{Inflow} + \text{forecasted Local Inflow}) \quad \textbf{(EQ 5)}$$

For the purpose of Equation 5, *Inflow* includes the value of the Inflow slot (at the time of the last dispatch) and the additional inflow resulting from the hypothetical additional releases from upstream reservoirs [1]. It also contains the proposed flood control release hydrograph from the last pass of the flood control method.

Once all power reservoirs have been visited (in priority order), the HydropowerRelease() function returns to the calling rule. The result is a list of {slot,value,object} triplets,. The triplets take the form {powerReservoir.slot, value for this slot, powerReservoir}. The slots include the Additional Power Release, the reservoir Outflow (if non-zero), and depending on method selection, the Load (if the "Seasonal Load Time" method (**HERE (Objects.pdf, Section 17.1.37.7)**).

Where values for additional power releases are zero, the triplet {reservoir.Additional Power Release, 0.0, reservoir} will be returned, but no triplet for the reservoir's Outflow slot will be returned, because the value of the Outflow slot will not change as a direct result of this rule. (It may change as an indirect result, if an upstream reservoir makes an additional power release.) This avoids unnecessary dispatching when no additional power releases are made on a tributary.

The rule then sets these slots, if different, and the reservoir and all downstream objects dispatch.

3.10 Fish Release

During the hours of the day when power is not generated, water may be released to meet minimum fish flows called Fish Release or QFish. Minimum fish flows below the reservoir are specified on a monthly basis (12 values per year) and stored on a data object in a periodic slot. The release of water to meet a downstream demand is determined based on the available fish water in the reservoir, the other off peak releases that are occurring and the monthly minimum fish flow. The fish storage in the reservoir is accounted for each timestep. Fish water is either released or is lost due to evaporation. It is replenished through three methods: all inflow is shared, the accumulated local inflow is shared, or no inflow is shared. Also, anytime the pool exceeds the top of conservation pool, the fish storage is replenished.

Power releases are always made at full turbine capacity for as many hours of the day necessary to meet the load. The Fish Release is defined as the release of water that occurs when power releases are not being made. The Fish Release is considered a spill. To model this in RiverWare, a rule must be developed that calculates the additional water that must be released to meet the fish flow. Because this release is a minimum release, it is assumed that there will be no downstream flooding. Therefore, no additional functionality needs to be developed; the existing rule language is sufficient to simulate this release.

1.The notion of hypothetical releases is significant here because a rule function cannot have side-effects (i.e., change the state of a simulation object). This means that none of the computations made by the HydropowerRelease function are visible until/unless the Hydropower *rule* succeeds in setting all the slots returned from the HydropowerRelease function.

3.10.1 Accounting setup on Reservoir and Control Point objects

This model will use a simplified accounting network where each Reservoir has a storage account. Water will accumulate in the account and be released from the account according to user defined logic. The released accounting water will not propagate downstream. Following is a description of the modeling changes necessary by the user.

1. Enable accounting by selecting the “Enable Accounting” under the Accounting menu.
2. Switch the controller from Rulebased Simulation to Inline Rulebased Simulation and Accounting.
3. Create a new Fish storage account on each desired reservoir.
4. Create a new Fish pass-through account on the object below the reservoir. Create a supply link between the two objects. This is necessary so that the user can set the outflow from the Fish account using a rule. A sample setup is shown in the figure.

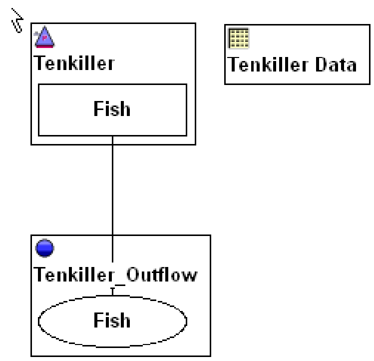


Figure 18. Sample Accounting Setup

5. Develop User Defined Accounting Methods that describe the logic used to bring water into the account through the Slot Inflow according to one of the following methods:

- All inflow is shared and the fish account is fully replenished when the pool exceeds the top of conservation pool.
- The accumulated local inflow is shared and the fish account is fully replenished when the pool exceeds the top of conservation pool.
- No inflow is shared but the fish account is fully replenished when the pool exceeds the top of conservation pool. A sample method that does this is shown in Figure 19.

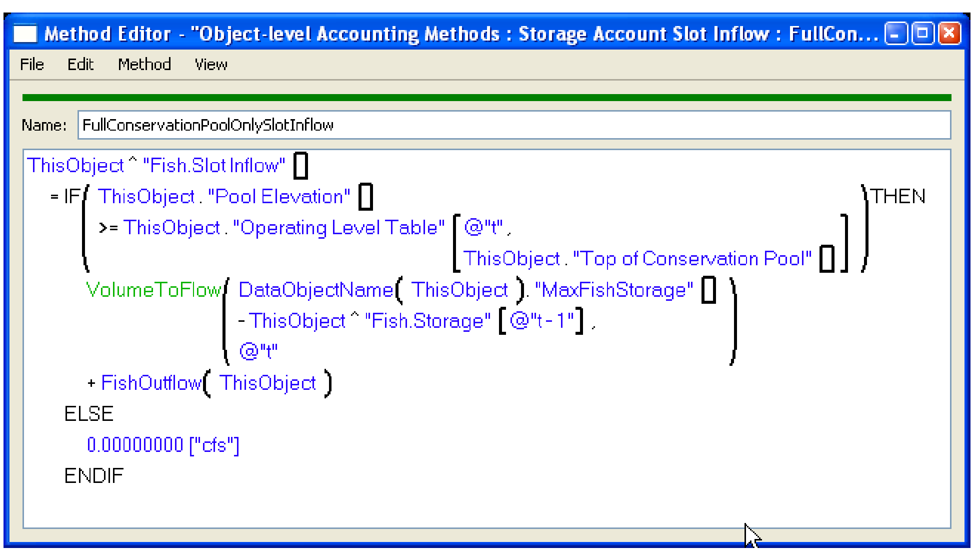


Figure 19. Sample Accounting Setup

6. Develop User Defined Accounting Methods that describe the logic used to account for Evaporation losses. Evaporation losses on any day is calculated as the current fish storage divided by the current total conservation storage times the current day total evaporation loss.

3.10.2 RPL Implementation

The rule should calculate the additional fish release necessary, set the **Outflow** slot and **Bypass** or **Regulated Spill** slot, by looking at the **Off Peak Spill** on the reservoir. This value is the water spilled (through regulated spill structures) during the non-peak (non-generating) hours of the day. If the **Off Peak Spill** does not meet the fish requirement, then the Reservoir outflow and bypass or regulated spill will be increased. To calculate the Fish Release, it is necessary to keep track of the volume of Fish water that is stored in the reservoir. This is best accomplished using the Accounting functionality in RiverWare. The accounting system in RiverWare provides a robust system of accounts and methods to keep track of different types of water in a system.The release rule should look at the previous Storage on the Fish account and available Fish Inflows to make sure that there is available water. When the reservoir dispatches again, the **Turbine Release**, **Peak Release** and **Peak time** should remain the same, only the **Off Peak Spill**, will change. A separate assignment will set the outflow supply from the Fish account equal to the additional water released from the reservoir. This assignment must set the outflow from the account and increase the outflow from the reservoir. Both of these outflows must be averages over the entire timestep but will actually be over the Off Peak period. Therefore, when writing the rules, it is important to calculate the correct value (off-peak) and then convert it to an average over the timestep. A sample rule is shown in Figure 20.

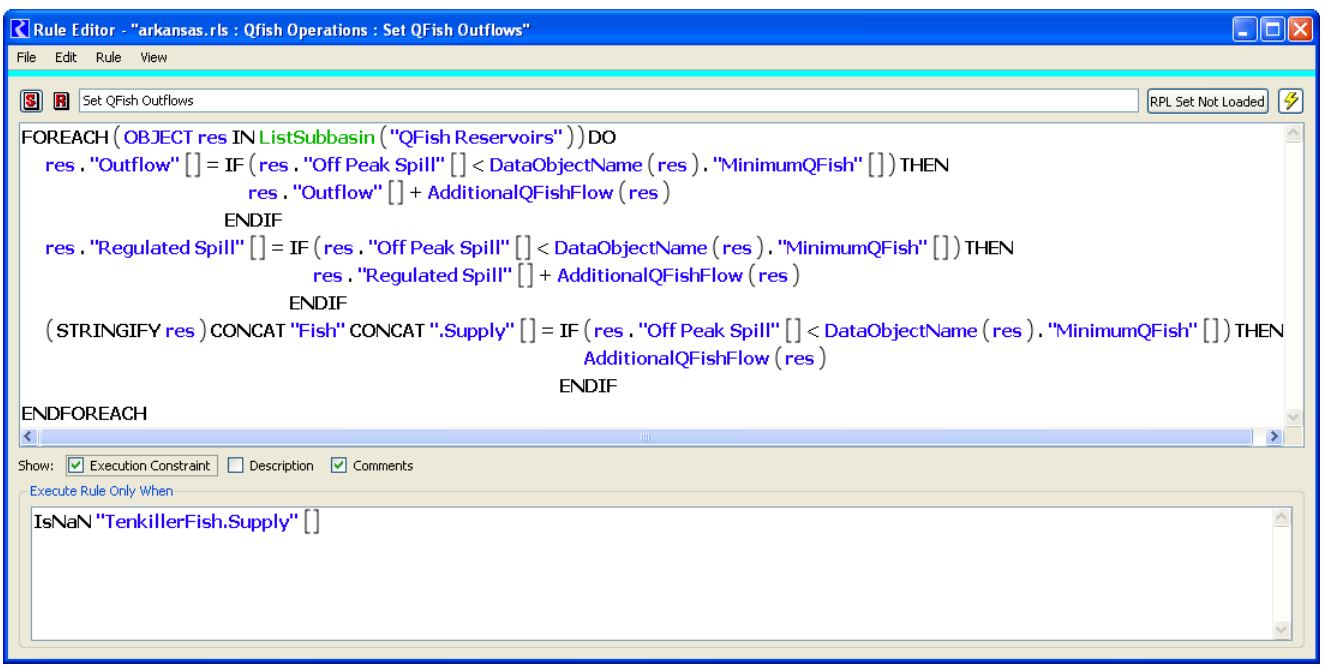


Figure 20. Sample Fish Release Rule

When this rule executes, the following is done:

1. A Fish rule fires and checks to see if the off-peak spill is enough to meet the fish requirement. If not met, the outflow is increased to meet the requirement limited by available fish water. Another assignment will set the outflow supply from the Fish account and the bypass or regulated spill slot.

2. If the outflow was reset, the reservoir will dispatch given inflow and outflow. The dispatch method will call the Peak Power Equation with Off Peak Spill method which calculates the tailwater elevation, Peak Release and peak time. Storage, Pool Elevation and the spill slots are set using the mass balance equations.

3. The reservoir's accounting system will solve for Fish slot inflow and gain loss according to user defined methods. The storage account will then solve for Fish account storage.

3.11 Direct from Reach Diversions

Water is diverted directly out of the river to meet demands. This operation does not require additional rules but does require additional model setup.

3.11.1 Model setup

A reach object must be added for each demand point. The reach will be set up to have a diversion and will have a Min Diversion Bypass method selected. The minimum base flow requirement will be input on the Minimum Diversion Bypass slot. The reach will be linked to a water user with the appropriate methods selected. The water user object will always divert the computed Diversion Request as long as there is sufficient water available. If not, it will take the available water. The available water is computed as the total flow in the reach (Inflow) minus the Minimum Diversion Bypass.

Make sure to also add these reach objects to the subbasins created for other operations.

3.11.1.1 Reach methods

On the diversion reach, select the following methods:

Category	Method	Link
Routing	No Routing	HERE (Objects.pdf, Section 22.1.1.1)
Local Inflow and Solution Direction	No Local Inflow, Solve Outflow	HERE (Objects.pdf, Section 22.1.3.7)
Diversion from Reach	Available Flow Based Diversion	HERE (Objects.pdf, Section 22.1.17.2)
Min Diversion Bypass	Periodic Min Bypass	HERE (Objects.pdf, Section 22.1.18.5)

3.11.1.2 Water User and/or Diversion Object methods

Either a water user or a diversion object can be used, but typically a water user is recommended. On the water user, the following method selections are recommended:

Category	Method	Link
Return Flow	**Fractional Return Flow**	**HERE (Objects.pdf, Section 27.1.4.2)**
Fraction Return Flow Input	**Input Fraction**, **Zero Fraction**, or **Periodic Fraction**	**HERE (Objects.pdf, Section 27.1.5)**
Diversion and Depletion Request	use either **Input Request**, **Periodic Diversion Requests**, or **Reservoir Level Lookup**.The last one allows the Diversion Request to be based on the operating level of a reservoir	**HERE (Objects.pdf, Section 27.1.1)**

Note: If diversion requests are based on a reservoir level, but the reservoir level is disabled because it is set to Pass Inflows, requests are set to zero. See the note **HERE (Section 4.3.1)** for more information.

USACE-SWD Misc

4. Unregulated Conditions

Unregulated flows are defined as those flows that would have occurred without the dams/reservoirs regulating the river. Other terms for these flows are unregulated, uncontrolled, pre-project, and unconstrained flows. The unregulated flows are often used to compute the system “Benefits”, that is, how much benefit has been gained by constructing the projects.

There are two steps to compute the unregulated flows:

- Select methods to remove the effect of one or more reservoirs. When specified, the reservoir will pass inflows.
- Run the model with the reservoirs removed and take snapshots of the desired slots.

Scripts, as documented **HERE (ScriptManagement.pdf, Section 1)**, can be used to automate this process.

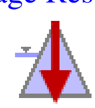
In the following section, we first describe how to disable the reservoir. In section 4.2, we describe how to make the runs for the regulated, fully unregulated, and partially unregulated systems.

4.1 Functionality to remove the effect of a reservoir

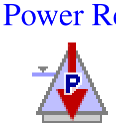
To remove the effect of reservoir storage and other processes, select the **Pass Inflows** method in the **Disable Reservoir Processes** category as described **HERE (Objects.pdf, Section 24.1.30.2)**. This method does two things:

- Changes the Workspace Icon: When this method is selected, the icon on the workspace is modified to indicate the Reservoir is not calculating storage as shown to the right.
- Enables an alternative dispatch method: When this method is selected, ONLY the **Outflow Equals Sum of Inflows** dispatch method is available. This dispatch method sets the Outflow = Σ Inflows.

Storage Reservoir



Level Power Reservoir



4.2 Model Run Mechanism

To compute both regulated and unregulated flows, two simulations are required. This section describes an approach for USACE-SWD:

- **Computation of flows for a fully unregulated system**: any model can be configure to compute unregulated flows by simulating the system without the effects of the reservoirs in a separate run using scripts. Snapshots will be used to compare results.

- **Computation of flows for a partially unregulated system**: a subset of the reservoirs can be disabled to represent a partially unregulated system. This will allow the user to make a run with the partially regulated system using Rulebased Simulation. Snapshots will be used to compare results.

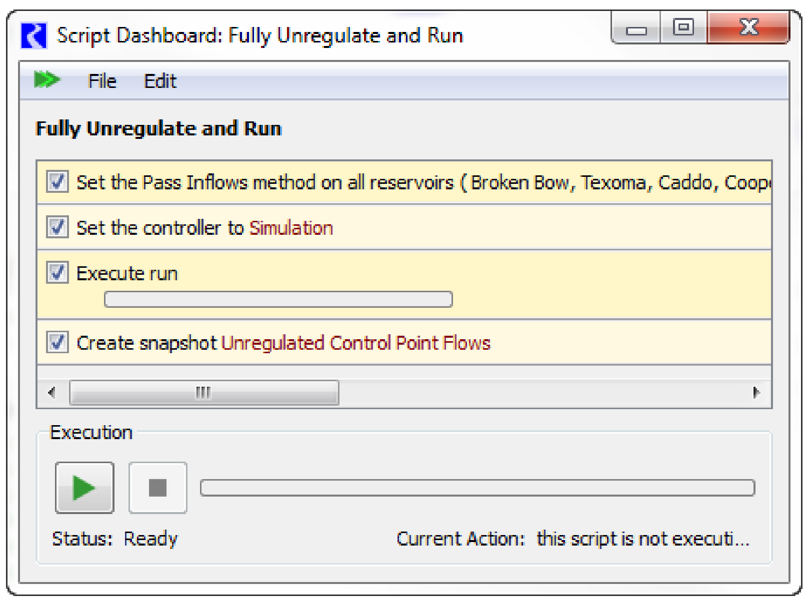
Although these two applications are similar and use the same approach, they will use different scripts. The fully unregulated system is essentially a pure simulation with no rules needed. For the partially unregulated system, rules are necessary and those rules and the flood control algorithm must know how to deal with a disabled reservoir. The partially unregulated run requires the user to select or describe what they want to model, so is more complex and warrants a different script.

The following sections describe the design for these two approaches

4.2.1 Computation of flows for a Fully Unregulated system

When you want to compute the flows at any point in the system without the effect of any reservoir, two simulations will be run. The first run computes the unregulated flows, the second computes the standard regulated flows using rulebased simulation. Results can be analyzed using snapshots. Simple scripts will 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:

- Sets a method on each reservoir indicating it should Pass Inflows as described above.
- Sets the run controller to **Simulation**
- Runs the model. With the new method, the reservoirs will still forecast hydrologic inflows, but no other physical processes will be 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:

- Sets the method on each reservoir to its normal regulated mode (**None** method in the **Disable Reservoir Processes** category).
- Sets the run control to **Rulebased Simulation**

- Loads the RBS ruleset set if necessary
- Runs 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.

4.2.2 Computation of a flows for a Partially Unregulated system

Computation of flows for a partially unregulated system will be performed in a separate run and results can be analyzed using snapshots. Again, scripts will be used to modify the model and make the run. Following is the conceptual approach for a POR model:

1. Execute a script which does the following to "Unregulate" specific reservoirs:

- Sets a method on one or more reservoirs indicating it is Passing Inflows as described in section 4.1.
- Disables any rules that reference the reservoir (In particular, the Surcharge rule for the disabled reservoir **MUST** be disabled).
- Run the model in rulebased simulation. The unregulated reservoir will still do its hydrologic inflow forecasting, but no other physical processes. When it dispatches, it will set outflow = ΣInflows, thus propagating the flows downstream.
- Take Snapshots of key results to preserve their values.

2. Execute a script which does the following to undo the script above and impose the "Regulation" again:

- Set a method on the reservoir indicating it is back to its normal "Regulated".
- Enable any rules that reference the reservoir (Surcharge).
- Make a run model

3. Compare the resulting flows with the snapshots from the unregulated run.

4.3 Information on other objects when reservoirs are disabled

Because the USACE-SWD methods are a system approach, disabling reservoirs can affect other objects, processes, or operating policies. The following sections identify some issues to keep in mind when using this functionality.

4.3.1 Diversions based on Disabled Reservoirs

No diversions are ever made from a disabled reservoir. Reach diversions will continue to be made based on their diversion requests.

On the Water User, and Diversion Object, there are methods called Reservoir Level Lookup which compute the Diversion Requested based on the level of a single reference reservoir. If that reference reservoir is disabled, the request slots are set to zero and a brown warning message is issued (one per run). If you wish to use an alternative diversion request, add an action to the script to change the method. When you want to undo the unregulated conditions, that script could reset the method back to the original method and it would be re-regulated again. No manual changes would be necessary.

To turn off reach diversions or to use an alternative value, select a different request method on the water user or diversion object within the script that unregulates the system. The most likely choice would be the Periodic Diversion Request method which allows you to specify a periodic slot of diversion requests. A single value in the table would represent a scalar request. To undo this action, use the Set Method action to re-select the original method. The original table data will then be used. Alternatively, set the Diversion to a specified value. Use the Set Series Slot Values action to set the Diversion equal to a set of values for all timesteps in the run. To undo this action, add a Set Series Slot Flags action that changes all those slots zero diversion slots to have the O flag. On the next run, the values will be cleared and the values will be re-computed. This approach would be useful if you have diversions that are not periodic or scalar.

In addition, in the Reservoir Diversions policy using the approach described **HERE (Section 3.8)**, you can divert water from one reservoir to another reservoir. That “Demand” reservoir can limit the diversion from the supply reservoir. If the Demand reservoir is disabled and set to Pass Inflows, it will issue an error that it does not have an operating level. You should change the method on water user to not limit by reservoir level.

4.3.2 Seepage methods

In some models, Seepage is set to an input and linked to the downstream object. If this was used this way in an unregulated model, it would “add” water to the system. Instead a seepage method like **Single Seepage Value** should be selected on the disabled reservoir. Then no Seepage will be set.

4.3.3 Hydropower releases downstream search

The hydropower release algorithm described **HERE (Section 3.9)** proposes a hydropower release and then searches downstream to see if it causes additional downstream flooding. This algorithm is designed to stop at a downstream reservoir. But, if the downstream reservoir is disabled and is set to Pass Inflows, the search continues downstream as though the reservoir was not there.

4.3.4 Low Flow Requirements based on Disabled Reservoirs

On the Control Point, there is a method called **Reservoir Level Lookup** which computes the low flow requirement based on the level of a single reference reservoir. If that reference reservoir is disabled, the following behavior now occurs:

- Simulation: (fully unregulated), no Low Flow Requirement is computed, but the run can continue. This assumes no operations are made to meet these demands.

- Rulebased Simulation: (partially unregulated), an error is issued as the low flow computation is not valid. A different Low Flow Requirement method should be selected.

4.4 Benefits computations

The flood reduction and benefits computation can be made using a combination of rules, expression slots, and/or statistical slots using regulated flows and unregulated flows.

5. Plotting, Output, and Analysis

Once a run has been made, there is typically a large volume of data to analyze. This section provides an overview (and links to additional documentation) of some of the features available to view, analyze, an debug model results. This includes plotting, statistical slots, SCTs, expression slots, and the model run analysis tool. The model run analysis tool includes a special results section which includes information specific to USACE methods.

5.1 Plotting

The RiverWare plotting utility allows the user to plot series, tables, contours, parametric, and periodic curves. This utility is described in detail in **HERE (Output.pdf, Section 2)**. In the plot utility, you can quickly choose to plot either all the data in a slot , only data in the run range , or only data in a specified range .

Saved plots configurations can be moved amongst models by exporting the configuration from one model and then importing into another. This utility is described **HERE (Output.pdf, Section 1.2)**.

5.2 Statistical Slots and Probability plots

Statistical Table slots compute statistics on a series of data. The user creates the slots and then selects the series slot for which the analysis is to be performed. Then, the user selects the statistical function to use and applies data filters as necessary. (Make sure to configure the statistical slots to only analyze data from the run range, not data before the start timestep or after the finish timestep. To do this, in the filter, choose the "Run Range" option) The result is a table of data that can be viewed, plotted, or exported. Statistical slots that result in data that is decimal or percent (i.e. an exceedence curve) can be plotted with a probability scale. This is similar to plotting the data on probability paper.

Statistical slots are described **HERE (Slots.pdf, Section 4.11)**.

5.3 SCT

Another useful tool for viewing data is the System Control Table (SCT). The SCT is a customizable, editable "spreadsheet" view of slots in a RiverWare model. The SCT presents series data in a scrollable grid of numeric values and lists other types of slots for easy access. The SCT is a view into the model and as such, the user can have more than one SCT for a given model. For example, the user may create (and save) an SCT that shows all reservoir Outflows or one for all reservoir Storages. Or, the user may configure an SCT to show all of the desired information about a single reservoir.

The SCT is described in detail **HERE (SCT.pdf, Section 1)**.

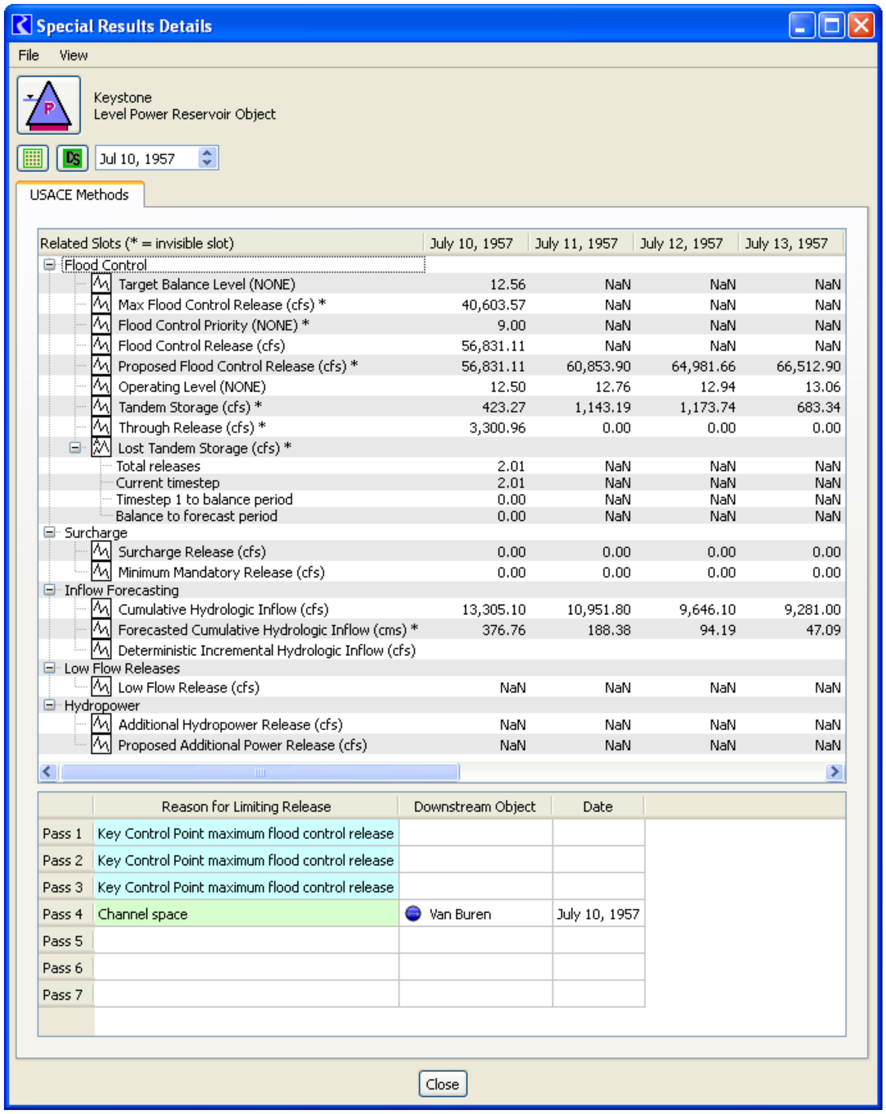
5.4 Expression Slots

Expression slots are computational expressions that can contain slot names and other variables. The user can either create a **Series Slot with Expression** or a **Scalar Slot with Expression**. They are used to calculate any quantity the user wishes to see. For example, the user may create slots that compute "combined Storage of all Reservoirs," "weekly average Outflow of Reservoir 1," "the ratio of Hydrologic Inflow to Control Point Inflow." The logic for an expression slot is developed in the RiverWare Policy Language (RPL). Expression slots can be evaluated automatically at the beginning of run, beginning of timestep, end of timestep or end of run. They can also be evaluated outside of a run.

Therefore, there is a lot of flexibility with expression slots and are very useful for debugging, post processing data, and other analysis. Expression slots are described **HERE (Slots.pdf, Section 4.6)**.

5.5 Model Run Analysis - Special Results Details Dialog

The **Model Run Analysis** tool is a grid showing how each object solved on each timestep. This utility is described in general **HERE (ModelRunAnalysis.pdf, Section 2)**. The **Special Results Details** dialog provides additional information for each object/timestep including information specific to USACE methods as shown to the right. This section provides more information on accessing and using the tool.



5.5.1 Accessing and Navigation

Access the **Special Results Details** dialog from the following locations:

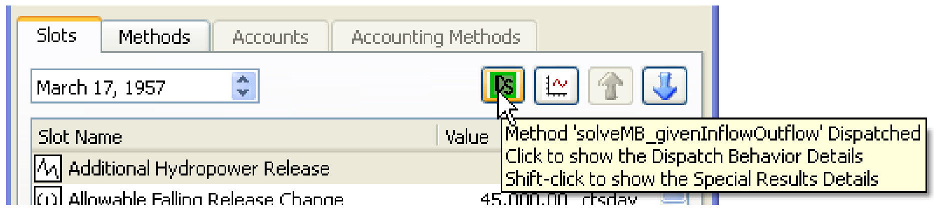
Model Run Analysis: From the model run analysis tool,

- Click on the **Details...** button and choose Special Results from the **Details:** pull-down menu. The

  details will be shown docked in the main **Model Run Analysis dialog**. Click the **Detach** button to use it as a separate dialog.
- On any cell in the **Model Run Analysis** dialog, right click and choose **Show Special Results Details...** to show the details in a new window.

Open Object Dialog: From the Open Object dialog,

- Click **View ➥ Special Results Details...** menu to show the details in a new window.



- Shift-click on the dispatch button to show the details in a new window. The button's tool tip summarizes the dispatch behavior but also lists the behavior when the button is clicked (open the **Dispatch Behavior Details** dialog) and when it is shift-clicked (open the **Special Results Details** dialog).

As with other model run details dialogs, **Special Results Details** dialogs can be undocked to display the information in a separate window. Because this dialog is naturally large due to the amount of information it displays, viewing it undocked is most likely preferred. Click the **Detach** button to undock and use it as a separate dialog. Use the **File ➥ Dock in Model Run Analysis** to re-dock it. When docked in the **Model Run Analysis** dialog, a splitter allows the dialog to be resized as necessary.

5.5.2 Layout and Overview

The **Special Results Details** dialog contains a tab labelled **USACE Methods** which presents information related to certain USACE methods described in this document. Only Reservoirs, Control Points, and Computational Subbasins have information that is presented by this dialog. Following is a screenshot and description of the tool.

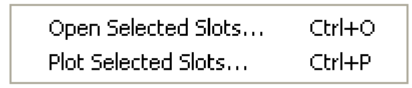
5.5.2.1 Related Slots

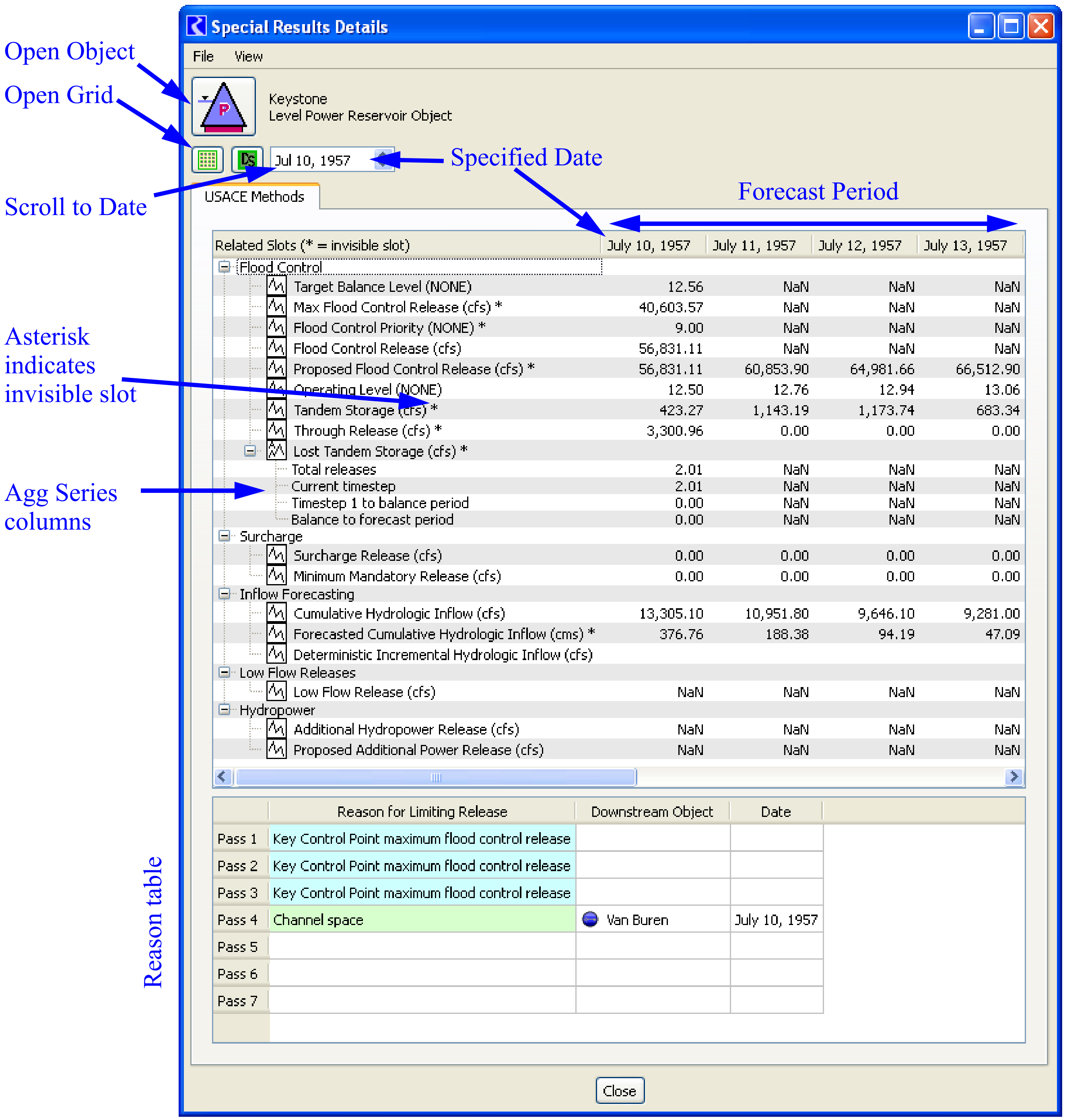
The **USACE Methods** tab displays results in two tables. The upper table displays the values of several series slots in the time range from the given timestep through the Forecast Period (the forecast period is taken from the computational subbasin which has a non-null Flood Control method selected, of which there should only be one).

The slots displayed are ordered into functionality groups: Flood Control, Surcharge, Inflow Forecasting, Low Flow Releases, and Hydropower. The displayed slots are those likely to be helpful in understanding the behavior of the various related USACE methods. This list of slots is based on the method selections and includes both intermediate results that are not normally visible to the user as well as regular slots that are closely related to the flood control algorithm. The slots are shown in a tree-view so you can control the display by clicking on the + and - operations.

Note: Invisible slots are shown with an asterisk. This indicate they are not visible or accessible in other dialogs but are saved in the model file. These are slots that were “temporary” prior to version 6.1. This dialog is the only place to access these invisible slots. Display units are configured on the Unit Scheme. Also, you can create an exception to the scheme for this slot by opening the slot, then modify using the **View ➥ Configure** menu.

Any slot listed in this dialog (including the invisible ones) can be **opened** or **plotted** using right click context menus.





5.5.2.2 Reason For Limiting Release Table

The lower table displays diagnostic information on why the Flood Control release was limited. This table is only displayed on Reservoirs and is only populated when Operating Level Balancing is selected. Note that this table displays information that prior to 6.1 was encoded on the “Temp Reason” and “Temp Limiting Control Point” slots. Each row represents a pass of the flood control algorithm. Each time the flood control release is reduced by a constraint, the reason for the last reduction is written. On dates that have few or no passes, some rows will not have values. While diagnostics might refer to other forecast timestep, only the **first** forecast timestep’s reason (i.e. current timestep) is saved here. The Downstream Object and Date columns will have values if the reason is “Channel Space”. Otherwise they will be blank.

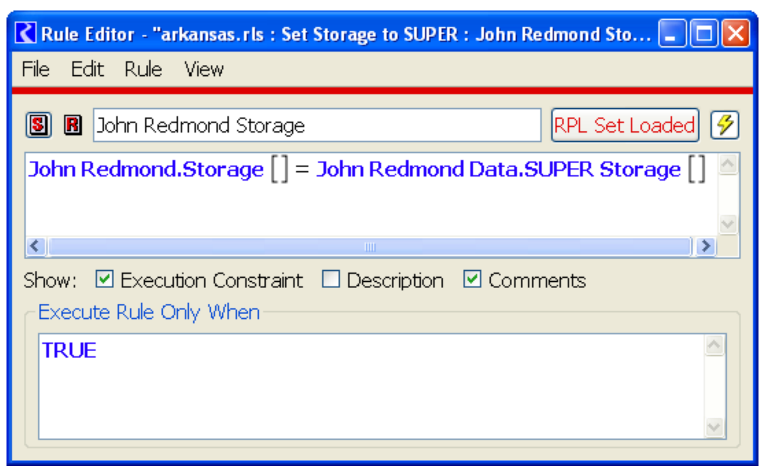
Diagnostic String	Discussion
Spillway constraint	The spillway cannot accommodate any more outflow. This constraint applies to all outflows, including surcharge releases, through releases, and this object’s flood control release.
Stepped hydrograph smoothing	The release was limited by the first release in a stepped-down hydrograph that releases the *entire proposed* flood pool volume over the rest of the forecast period with an Allowable Falling Release Change reduction at each timestep. This constraint is applied once, to the first timestep only.
Channel space	Channel space at a downstream control point limited the release. The value in the Downstream Object column indicates which downstream control point is the cause and at which forecast Date. This information indicates which downstream control point is lacking in adequate channel space and limited a flood control release on each pass of the flood control algorithm.
No increase after timestep 0	On its last pass, the algorithm does not allow an increase in release over the prior pass’s proposed release for any timestep after the first timestep in the release hydrograph. This string will show up in diagnostics, but not in the slot, since the slot contains only those reasons that apply to the first timestep of the forecast period.
Key Control Point maximum flood control release	The Maximum Flood Control Release slot value (computed by a key control point) limited the release.
Volume above key Control Point balance level	The volume intended to be released is that found to be above the balance level assigned by a key control point at the end of the balance period. This volume is reduced by each scheduled release in the forecast period, and the remaining amount limits subsequent scheduled releases.
Volume in flood pool	The flood control release was limited by the total volume forecast to be in the flood pool at the time this reservoir is trying to make a release (this takes into account tandem storages and forecast flood control releases).

Diagnostic String	Discussion
Allowable falling release change	The release was limited by the first release in a stepped-down hydrograph that releases the *remaining proposed* flood pool volume over the rest of the forecast period with an Allowable Falling Release Change reduction at each timestep. This constraint is applied to each timestep in the forecast period.
Allowable rising release change	The release was limited by the Allowable Rising Release Change added to the prior timestep's release.

5.6 Comparing to SUPER

It is recommended that the user test the model at each stage of development. For example, the surcharge release results should be verified before moving on to flood control. The following testing strategies explain how to compare RiverWare results with results from the SUPER model. However, a similar testing strategy may be devised to compare RiverWare with results from another model or historical data.

5.6.1 Set Storage Rules



For testing purposes, it may be useful to create rules that set the storage slot on each reservoir to the final, known storage value taken from another model or from historical data. These rules can be used to incrementally test portions of the model. Since these rules need to overwrite the values computed by the surcharge release and flood control rules, they must be higher priority. Add a policy group above the flood control policy group and add a rule for each reservoir in the model. The rules should be ordered such that the upstream reservoirs execute first and the downstream reservoirs execute last (same order as the surcharge release rules). Each rule should set the reservoir storage slot equal to the final, known storage value (either taken from historical data or another model run). These values should be stored in a custom slot in RiverWare. An example rule is shown below. These rules should be used for testing purposes only and would not be a part of the final flood control model.

When these storage rules are active, they will overwrite the results of the policy. Each rule will reset the reservoir storage to the known values (stored in a custom slot). This will trigger each reservoir to re-dispatch with the solveMB_givenInflowStorage method. Each reservoir will compute a new outflow which will be routed downstream.

5.6.2 Testing strategies

The set storage rules, discussed above, are required to accomplish this incremental testing. These rules allow the RiverWare model to be reset, at the end of each timestep, to a state equivalent to the SUPER model at that timestep. This ensures that the RiverWare model begins the next timestep in the same state as the SUPER model. As discussed above, the set storage rules should be the highest priority rules and will reset the reservoir storage to the SUPER storage value for that timestep. The SUPER storage values need to be obtained from a SUPER run that only has the desired calculations enabled. Store these values in RiverWare on custom slots.

5.6.2.1 Testing Surcharge Release

In order to test the surcharge release results, the surcharge release rules and set storage rules need to be active. The regulation discharge and flood control policy groups should be turned off. The goal of this testing is to compare the surcharge releases computed in RiverWare with the surcharge releases computed in SUPER. Both models should produce the same results, within some accepted tolerance. The set storage rules are necessary because, without them, the reservoirs in RiverWare will fill up to the top of the flood pool and stay there (because only the surcharge release rule would be active). The user may be wondering why the flood control rules couldn't be used instead of the set storage rules. The reason is that it would defeat the purpose of incremental testing. If the RiverWare flood control results deviate from SUPER, then the models have deviated and it would be meaningless to compare surcharge release results.

When the RiverWare model is run with the surcharge release and set storage rules, the model will compute the surcharge releases at each timestep. Then the set storage rules will reset the reservoir storages to the known SUPER storage values. The final surcharge release values can then be compared to the final SUPER surcharge release values. If the values do not compare, debug the problem. When these values match up, proceed to the next level of testing.

5.6.2.2 Testing Regulation Discharge

To test the regulation discharge results, the surcharge release, regulation discharge, and set storage rules should be active. The regulation discharge and empty space results computed by RiverWare should compare with the SUPER results.

5.6.2.3 Testing Flood Control Results

Once the surcharge release and regulation discharge calculations have been verified, the flood control calculations can be tested. To test these, the surcharge release, regulation discharge, and flood control rules should be active. The set storage rules should NOT be active because they would overwrite the flood control results (although some types of specific testing may need to make use of the set storage rules).

5.6.2.4 Testing of Low Flow Releases, Reservoir Diversions, and Hydropower Releases

Follow a similar approach as described above to test these three policies. The user should make a determination on a case by case basis as to whether the set storage rules should be enabled.

6. DMI's to import/export data

The Data Management Interface (DMI) can be used to import or export data from a model to an external data repository or database. There are two types of DMIs:

- Control File/Executable: provide a link to any external database or repository
- Database DMI's: provide an automated connection to HEC-DSS and HDB.

The USACE-SWD uses DSS as it data repository, so this section will focus on the use of the Database DMI connection to DSS. In USACE-SWD models, DMIs are used to import:

- Table and scalar data used to define the model
- Initial values
- Time series data representing uncontrolled area flows, evaporation, power demands, etc.
- Observed or existing data used for comparison

After a run is made, DMIs are used to export data to the DSS file to store the results of the run. The following section provides links to the DMI interface, the utility to record invocations of each DMI and then clear the values set by the DMI, and documentation on the interaction of RiverWare and CWMS.

6.1 Database DMI - DSS interface

The Database DMI interface allows the user to configure Name Maps and Datasets, and then define a Database DMI that uses them. The database DMI interface is described in detail **HERE (DMI.pdf, Section 5.1)** and will not be repeated here.

6.2 DMI Invocations and Clearing Input DMI values

The USACE-SWD uses DMIs to bring large volumes of table and series data into the model. Frequently, a previously developed model will be used for a study, but all of the data will be replaced with new data. To prevent using any of the previous data, a utility was developed to allow the user to clear out values set by a DMI. This utility has two parts, recording an invocation of the DMI and then clearing values set during that invocation. Following is an overview of the process a user would take. This utility is documented in detail **HERE (DMI.pdf, Section 2.4)**.

1. Within the DMI manager, the user configures that an input DMI should record invocations. With this toggle enabled, the DMI will record every time the DMI is invoked and the slots set.

2. The user executes the input DMI and the invocation is recorded. Also, with invocations enabled, all of the data that is set is given the “Z” flag instead of the “I” flag. Note, the Z flag behaves identically to the I flag.The user then performs the run as usual.

3. When the user decides that they want to start a new study with new data, the user opens the Invocation Manager and selects the appropriate invocation. The user then clicks on the **Edit ➥ Clear Selected Values** menu and any data that was set by that DMI is cleared out (i.e. set to NaN).

4. In this way, the user can go through all of the input DMIs and clear out any value that was set by a DMI.

To summarize, to clear out all imported data in the model, the user should make sure that all DMI’s have the Record Invocations toggle enabled. Then, when it is time to clear out the data the Invocation Manager can be used.

7. Yield Study

This document provides instructions to convert an existing working USACE-SWD model into a yield study using iterative MRM. A yield study is used to determine the largest average diversion that can be made from a single reservoir such that the reservoir will not drop below the bottom of conservation pool at any time during the run period. Other operating policies like Surcharge, Regulation Discharge, Flood Control, Low Flow Release, and Hydropower can be included in this analysis. The following files are needed for this analysis (obtain from riverware-support@colorado.edu):

- YieldStudy.rls.gz - An MRM ruleset that controls the run iteration and determines the yield for each run.
- YieldStudy.obj - An importable data object containing the slots referenced by YieldStudy.rls.gz
- YieldStudy.sct - An optional SCT used to view the integer indexed slots in the data object imported from YieldStudy.obj

Note: These files were updated for RiverWare 6.7, Please contact riverware-support@colorado.edu for the files that correspond to this document.

7.1 Steps

Given a working model for a basin, to conduct a reservoir yield study, the following changes need to be made:

1. Import the yield study data object.

- Select the workspace menu item **Workspace ➥ Objects ➥ Import Objects**... and import the file `YieldStudy.obj`.This will create a new data object named **Yield Study Data**.
- The slots on this object used for input data are as follows:

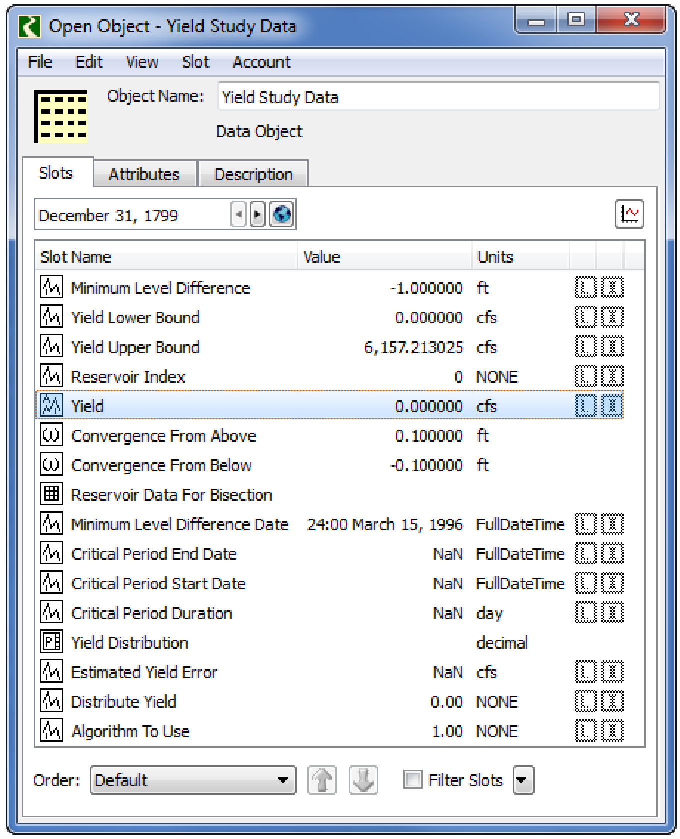


Figure 21. Screenshot of Yield Study Data Object

  - **Reservoir Data for Bisection**: The name of the reservoirs which are participating in the yield study should appear as the row labels of the **Reservoir Data for Bisection** Table Slot. Append rows as necessary using the Edit menu. Edit the row label as appropriate using the **View ➥ Edit Row Labels** menu. When editing row Labels, there is a useful button in the Edit Row Labels dialog to **Set Label(s) to an Object Name**. The yield study ruleset requires a lower limit on the yield for each individual reservoir. By default the algorithm will use 0.0cfs as the lower limit. If this is not a reasonable lower limit, you can provide an alternative value in the **Minimum Yield** column of the table. The yield study ruleset also requires an upper limit on the yield for each individual reservoir. By default it will use the average inflow, over the run, for each reservoir. If this is not a reasonable upper limit, you can provide an alternative value in the **Maximum Yield** column of the table. For certain algorithms, an **Initial Yield to Try** can be provided as a column.
  - **Convergence From Above, Convergence From Below**: The scalar values representing how accurate the yield value should be above or below the exact answer: **Convergence From Above** and **Convergence From Below.**
  - **Yield Distribution:** This optional periodic slot is used to specify factors for distributing the average yield throughout the year. The values in the slot are the fraction or percentage of the average yield to be applied to each timestep in the run. The values in this slot should average to 1.0 over the course of a year. A pre-run rule will check that they average to 1.0 and stop the run if they do not. If this slot is to be used, makes sure to use the MRM rule **Use Distribution Pattern?**. See item number 4 below.
- The remaining slots on this data object contain output values from the yield study. Most of them are indexed by the MRM run number.
  - **Yield**: Integer indexed series slot contains the results of the analysis (indexed by the MRM run number). Append columns as necessary using the Edit menu. On the “Yield” slot, modify the column label (using the “View ➥ Edit Column Labels” menu) to match the reservoir's name. When

editing column Labels, there is a useful button in the Edit Column Labels dialog to “Set Label(s) to an Object Name.”

- **Reservoir Index**: a integer indexed series slot that tracks which reservoir is being analyzed (indexed by the MRM run number). In a single reservoir study, this value will always be output as zero.
- **Yield Upper Bound, Yield Lower Bound**: Integer indexed series slots indicating the bounds used for the current iteration by the bisection method on the current reservoir (indexed by the MRM run number).
- **Minimum Level Difference:** Integer indexed series slot showing the minimum difference between pool elevation and the bottom of the conservation pool for the current reservoir at the current run (indexed by the MRM run number).
- **Minimum Level Difference Date**: Integer indexed series slot containing the date at which the Minimum Difference occurs.
- **Critical Period Start Date**, **Critical Period End Date**, and **Critical Period Duration**: These integer indexed slots have a unit type of DateTime. They are used to store the dates when the critical period starts, ends, and the duration, respectively.
- **Estimated Yield Error**: Integer Indexed series slot containing the underestimation error associated with the yield. This is calculated as the additional volume of water that must be released during the critical period, including an estimate of reduced evaporation, to lower the pool to the bottom of conservation pool on the critical date. More information on this calculation is provided **HERE (Section 7.3)**.
- **Distribute Yield:** Integer Indexed Series slot that keeps track of whether the user has configure to use a distribution pattern (value = 1) or a constant pattern (value = 0).
- **Algorithm To Use**: Integer Indexed Series slot that keeps track of whether the user has configure to use Bisection (value = 1), Heuristic A (value = 2) or Heuristic B (value = 3).

2. Disable diversions. For the purposes of the yield study, diversion is controlled by directly setting the reservoir Diversion slot values, so you will need to disable the Diversion and Water Users objects that would divert water from reservoirs participating in the yield study. This can be done in the Run Analysis dialog by selecting the object’s row and then selecting **Object ➥ Disable Dispatching**. This should only be done for the specific reservoirs for which you are finding the yield. The other reservoir’s Diversion and Water User objects could remain enabled as desired.

3. Load the standard rulebased simulation ruleset that represents the policy in the basin. Use the **Workspace ➥ Policy** menu. Disable aspects of the policy (ruleset) that should not apply for the yield study. In particular, if you do not want any of the reservoirs in the subbasin to divert water, disable the **Compute Reservoir Diversions** rule. If you want all non-yield reservoirs to divert as defined, then you will need to modify the standard ruleset. First, create an identical subbasin to the one specified in the Compute Reservoir Diversions rule. Then remove the reservoir, diversion and water user objects for which you are finding the yield. Use this new subbasin in the Compute Reservoir Diversion rule. This way, you will compute reservoir diversions for all reservoirs except those for which you are finding the yield. Other rules may or may not need to be disabled depending on the desired policies in place during the yield study.

4. Import the yield study iterative MRM ruleset YieldStudy.rls.gz.

- To import it, first open the MRM policy set by selecting the workspace menu item **Policy ➥ Iterative MRM Rules Set**.
- This will open the **Iterative MRM Rules**. To initiate the import into this set, select **File ➥ Import Set...**. Choose the file `YieldStudy.rls.gz`.
- This will open the import dialog which will display the policy group and utility group contained in the file; confirm that the items being imported do not conflict with existing names in your MRM policy and continue with the import.
- Set the Agenda Order of the MRM ruleset to “1,2,3,..”. This is done from the MRM Rules - Ruleset Editor **View ➥ Show Advanced Properties** menu. The Agenda Order is controlled by two toggles added to the bottom of the dialog.
- Specify which algorithm you wish to use. Turn on only one of either Rule 1, 2, or 3. Rule 1 specifies a bisection algorithm to find the solution. Rule 2 employs a combination bisection/heuristic search algorithm (Heuristic A) which uses the bisection method when the yield is too high and a heuristic approach when the yield is too low. In this latter case, it estimates the additional yield necessary to lower the reservoir to the conservation pool during the critical period. Rule 3 triggers a faster heuristic algorithm (Heuristic B) which tries the intermediate yield first and then uses the calculation after each successful fun. More information on these algorithms can be found **HERE (Section 7.3)**.

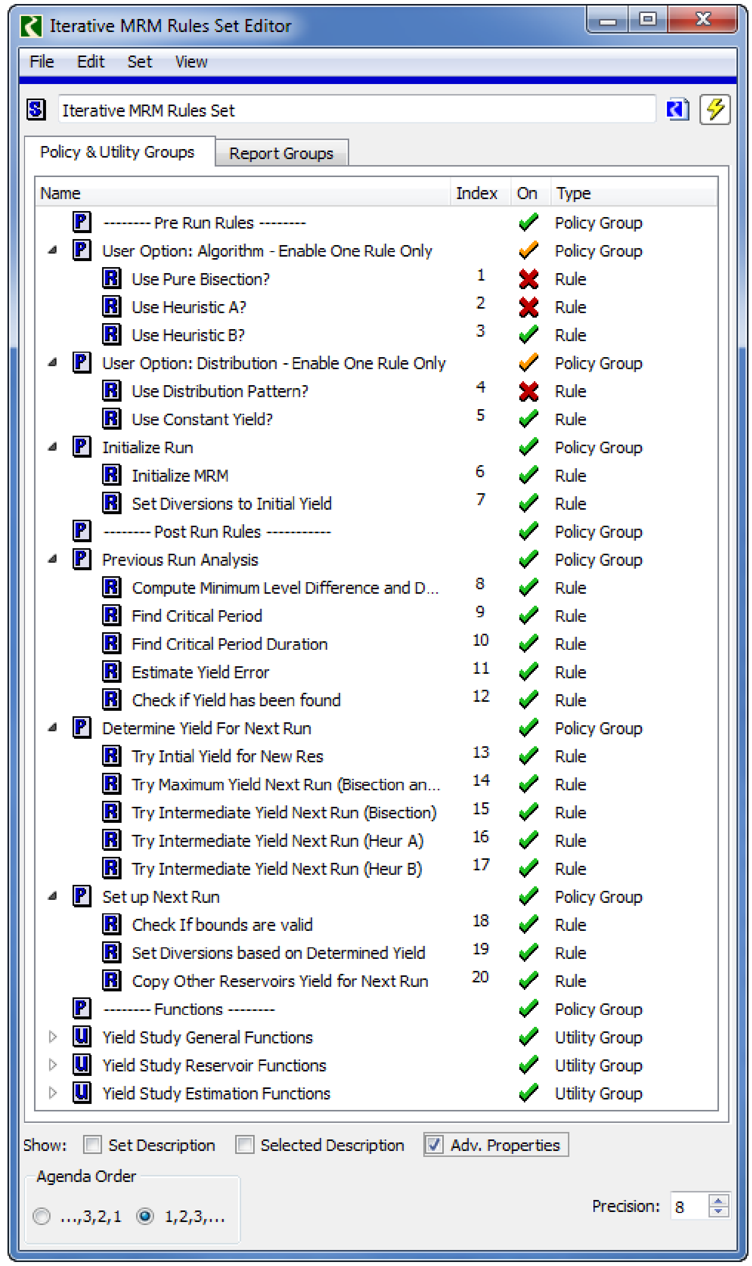
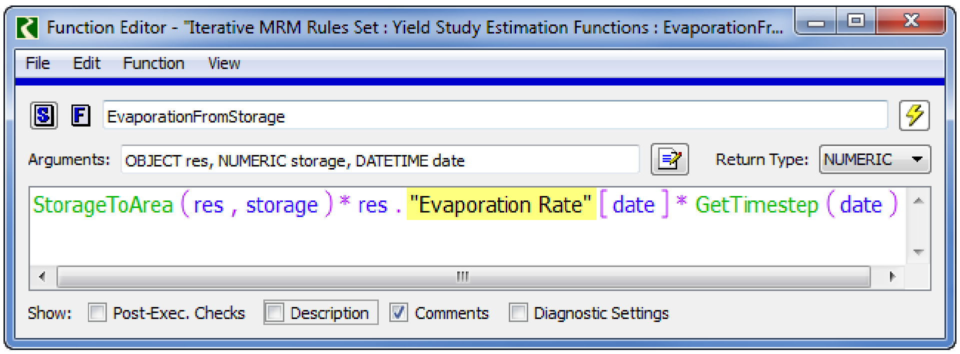


Figure 22. Screenshot of Yield Study Iterative MRM RPL set

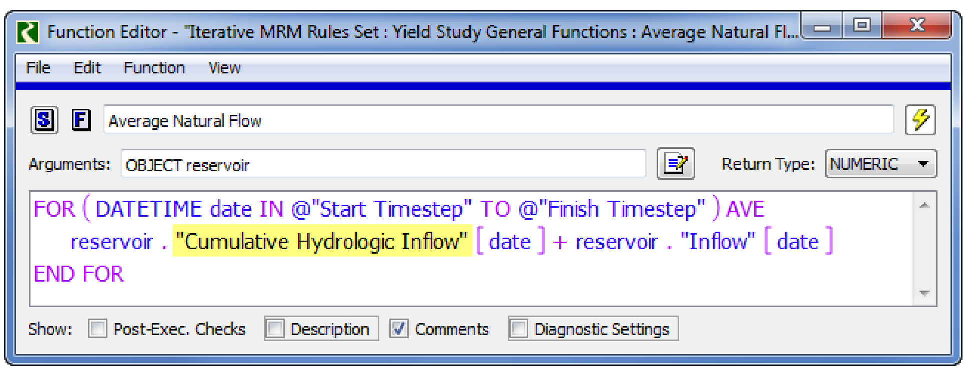
- Specify whether you want to use the average Yield or Distribute the Yield by enabling or disabling MRM rules 4 or 5. Rule 4 triggers a yearly distribution of yield. Rule 5 uses a constant yield.
- The RPL set should now look similar to Figure 22 with your combination of enabled and disabled rules. No other rules in the set need to be disabled.

- Following are two addtional changes you may need to make to the functions, depending on how the reservoirs are modeling inflows and evaporation.

Note: If you are using Heuristic A or B, you may need to change the function EvaporationFromStorage to access the correct Evaporation Rate slot. It should reference the slot with input data.



Note: If you are computing the maximum yield, you may need to change the function Average Natural Flow to access the correct hydrologic inflow slots. It should reference the slot with input data.



5. Create an MRM configuration for the yield study. More information on iterative MRM can be found **HERE (MRM.pdf, Section 4.3.3)**

- Open the Multiple Run Control Dialog by clicking on the appropriate workspace menu button or selecting **Control ➥ MRM Control Panel....**
- Create a new configuration MRM control dialog by selecting **Configuration ➥ New**.
- To edit the new configuration, double-click on its name in the MRM Run Control Dialog.
- This will open the MRM Configuration Dialog. Select Iterative mode and then select the **Iterative Runs** tab.
- In this tab, use the **Add** button to add the first seven (1-7) rules to the upper **Pre-Run Rules** box:
- Use the second **Add** button to add the remaining rules, 8-20, to the **Post-Run Rules** box.

- Select **Continue After Abort** and set Max Iterations to a number which is appropriate.
- Select that the **Pre-Run Rules** should execute **Before First Run**. Your screen should look similar to the following screenshot, Figure 23:
- Click OK to apply the changes and close the window.

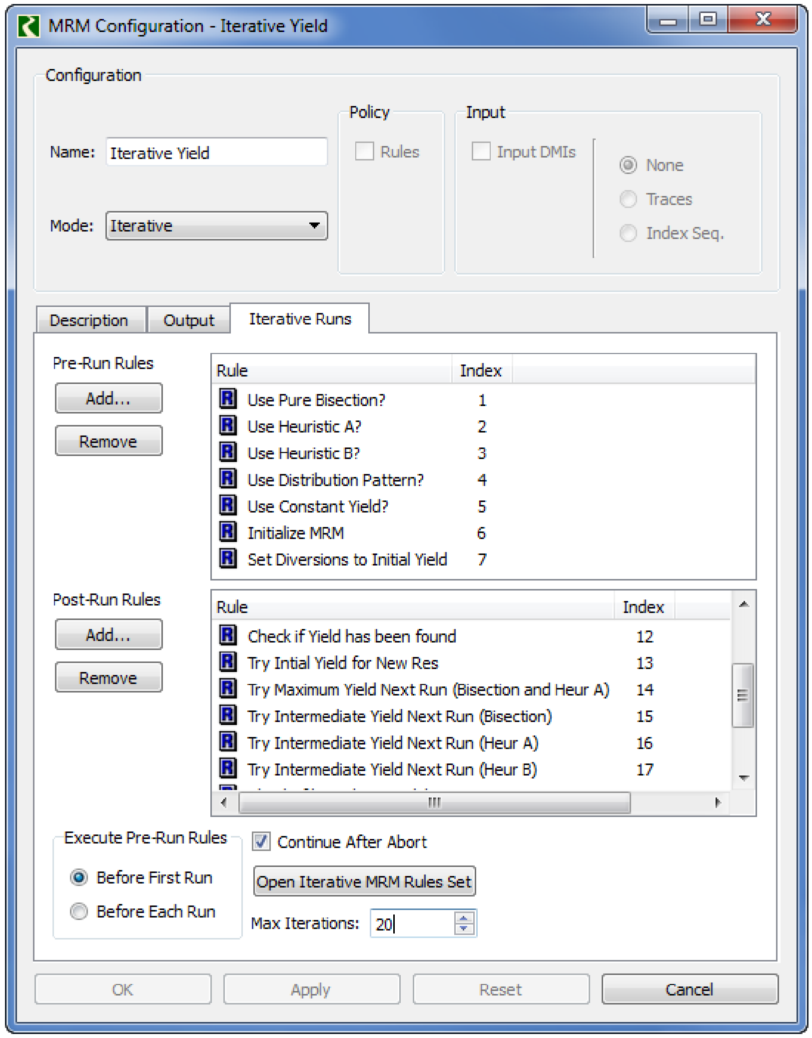


Figure 23. Screenshot of Iterative MRM Configuration

6. At any time before, during or after the run, you can open an SCT to see all the results of the run to that point. The SCT file is in YieldStudy.sct. If doing a multiple reservoir yield study, insert additional columns to show each column of the Yield agg series slot. The column labels of the SCT can be renamed using the Slots ➥ Set Label/Function menu. Make sure the SCT is unlocked or this option will not be available.

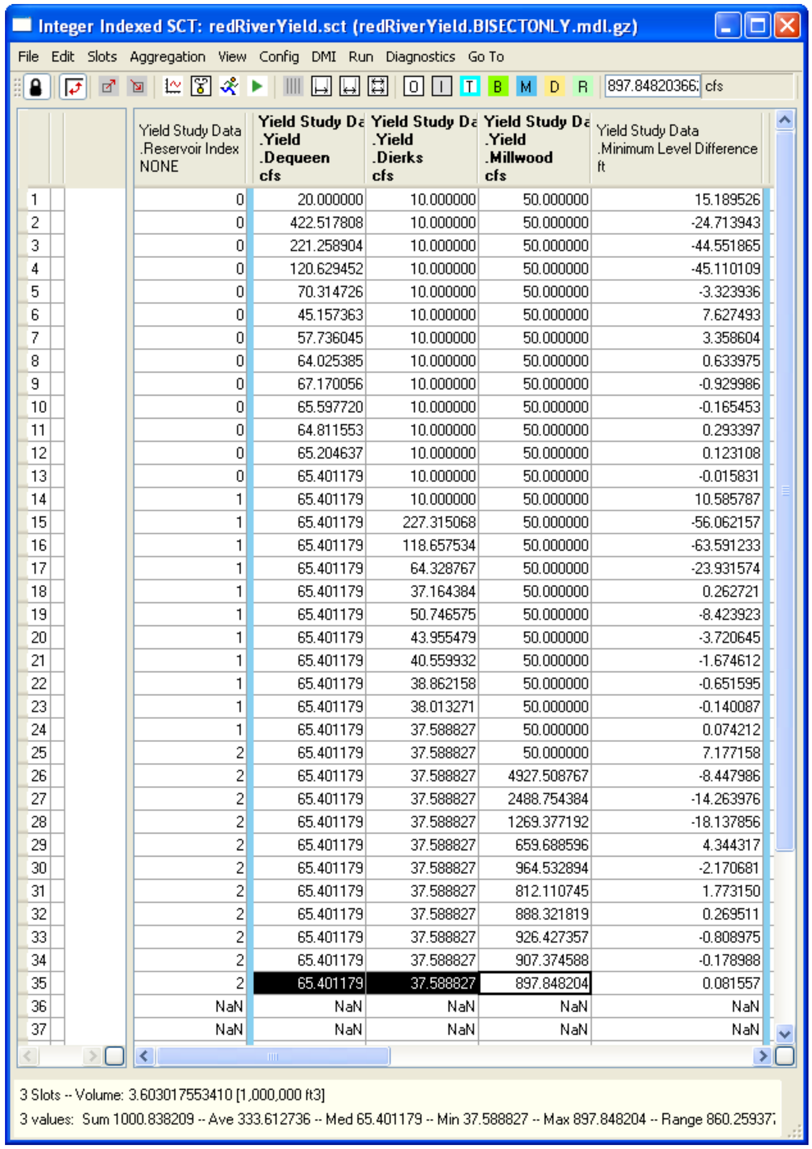
7. Conduct the MRM run as follows: In the Multiple Run Control, select the new MRM configuration on the Multiple Run Control dialog and click on the Run button.

8. After the run, the sequence of values tried for the individual reservoir yields may be found in the "Yield" slot of the "Yield Study Data" object. The easiest way to view all results is through the YieldStudy.sct.

7.2 Viewing Results

Sample results are shown in the following SCT screenshot, Figure 24. The Bisection method converged in 35 iterations. This section provides some guidance on viewing the SCT.

The SCT can be read as follows: Each row represents an iterative run. The average yield used for each run can be found in the Yield.Res column. For example, in the first run the minimum yield of 20 cfs was used for Dequeen, 10cfs for Dierks and 50 cfs for Millwood. In the next row, the maximum for Dequeen is used while the other two reservoirs continue to use the previous value. Then the yield for Dequeen is bisected. Only Dequeen's Yield is modified until it converges, then the algorithm moves on to Dierks. The Minimum Level Difference column shows the minimum difference between pool elevation and the bottom of the conservation pool. Note that convergence is met in rows 13, 24, and 35 where this value is less than 0.1ft. Note, the yields are average values, while the distributed values are set on the Reservoir.Diversion slot and can be viewed there. To determine the total yield, highlight each of the individual yields as shown and the sum is displayed at the bottom of the SCT.



	Yield Study Data .Reservoir Index NONE	Yield Study Da .Yield .Dequeen cfs	Yield Study Da .Yield .Dierks cfs	Yield Study Da .Yield .Millwood cfs	Yield Study Data .Minimum Level Difference ft
1	0	20.000000	10.000000	50.000000	15.189526
2	0	422.517808	10.000000	50.000000	-24.713943
3	0	221.258904	10.000000	50.000000	-44.551865
4	0	120.629452	10.000000	50.000000	-45.110109
5	0	70.314726	10.000000	50.000000	-3.323936
6	0	45.157363	10.000000	50.000000	7.627493
7	0	57.736045	10.000000	50.000000	3.358604
8	0	64.025385	10.000000	50.000000	0.633975
9	0	67.170056	10.000000	50.000000	-0.929986
10	0	65.597720	10.000000	50.000000	-0.165453
11	0	64.811553	10.000000	50.000000	0.293397
12	0	65.204637	10.000000	50.000000	0.123108
13	0	65.401179	10.000000	50.000000	-0.015831
14	1	65.401179	10.000000	50.000000	10.585787
15	1	65.401179	227.315068	50.000000	-56.062157
16	1	65.401179	118.657534	50.000000	-63.591233
17	1	65.401179	64.328767	50.000000	-23.931574
18	1	65.401179	37.164384	50.000000	0.262721
19	1	65.401179	50.746575	50.000000	-8.423923
20	1	65.401179	43.955479	50.000000	-3.720645
21	1	65.401179	40.559932	50.000000	-1.674612
22	1	65.401179	38.862158	50.000000	-0.651595
23	1	65.401179	38.013271	50.000000	-0.140087
24	1	65.401179	37.588827	50.000000	0.074212
25	2	65.401179	37.588827	50.000000	7.177158
26	2	65.401179	37.588827	4927.508767	-8.447986
27	2	65.401179	37.588827	2488.754384	-14.263976
28	2	65.401179	37.588827	1269.377192	-18.137856
29	2	65.401179	37.588827	659.688596	4.344317
30	2	65.401179	37.588827	964.532894	-2.170681
31	2	65.401179	37.588827	812.110745	1.773150
32	2	65.401179	37.588827	888.321819	0.269511
33	2	65.401179	37.588827	926.427357	-0.808975
34	2	65.401179	37.588827	907.374588	-0.178988
35	2	65.401179	37.588827	897.848204	0.081557
36	NaN	NaN	NaN	NaN	NaN
37	NaN	NaN	NaN	NaN	NaN

Figure 24. Screenshot of SCT used to view results from Yield Study

The SCT also contains (not shown) the bounds used in iteration and the critical period found for each run. At the end of the iterative run, the critical period can be found on the row in which that reservoir converged, i.e. 13, 24, and 35.

7.3 Algorithms

Following is a description of the available algorithms.

7.3.1 Bisection

The Bisection algorithm is the most basic and most robust algorithm implemented. All other algorithms fall back to the bisection if the individual run does not succeed. In Bisection, the minimum and maximum yield is tried first. Then the min and max are averaged or bisected to find the intermediate yield and a run is made with that value. After the third run, the algorithm decides if the yield is too high or too low and modifies the upper or lower bound as appropriates and bisects again. This procedure continues until the solution is found. It is robust as the min and max yields are tried first to bound the solution. No additional information on the reservoirs or physical processes are used, so it is often the simplest to get working. It can be slow as it uses a brute force approach; no information on the processes are used to estimate the next yield to try.

7.3.2 Heuristic A

In Heuristic A, first the minimum, then the maximum yields are tried. Then the average of those two yields is tried. If this third and subsequent yield estimate is too high, meaning the minimum level difference is negative, the bisection method will be used as described above. If the minimum level difference is positive the following approach will be taken to compute the next yield:

Assume that we have completed a simulation with a yield which resulted in a positive minimum level difference, i.e. the reservoir is above the bottom of conservation pool throughout the run. In this case, we have underestimated the yield. Examining the results from the simulation, we can find the dates of the *drawdown period* defined as the last date when the reservoir was full (at the top of the conservation pool) to the date of the minimum level difference. If we had guessed the correct yield then the drawdown period would have ended in a minimum level difference of zero, but we know that we underestimated the diversion because there is water remaining in the conservation pool at the end of the drawdown period. To exactly drain the conservation pool during the drawdown period, we would have needed to divert additional storage equal to the volume of the conservation pool at the minimum level difference date as well as enough to compensate for the reduced evaporation which would result from lower elevations over the drawdown period. Figure 25 shows a plot of storage versus time for this situation.

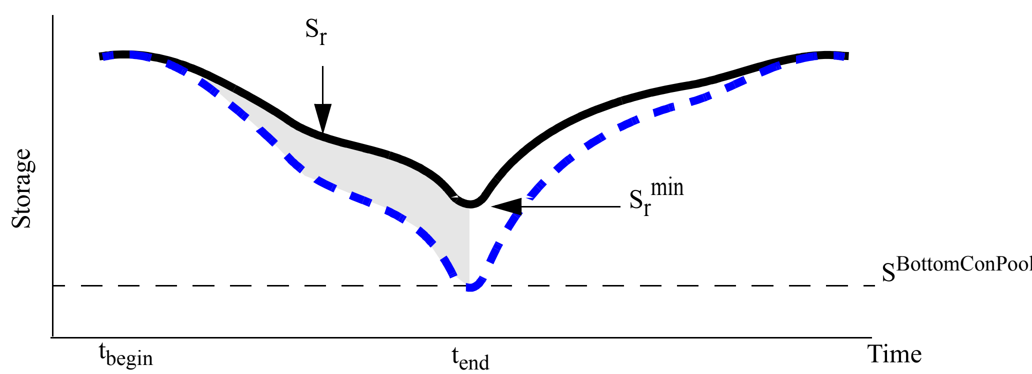


Figure 25. Plot showing a sample critical drawdown period. The curve S_r is the storage produced from the previous run. Additional water must be released to lower the pool to the bottom of conservation pool. The shaded area is the volume of water that must be released to scale the solid black curve down to reach the bottom of conservation pool. This would produce a curve similar to the dotted blue curve.

This analysis motivates the following equation for using the results from one run to choose a yield which will lead to a minimum level difference of zero in the next run:

$$y_{r+1} = y_r + \frac{E_r - \hat{E} + (S_r^{min} - S_r^{BottomConPool})}{\Delta t \cdot A} \quad \textbf{(EQ 6)}$$

where:

y = average yield

r = run number, r is the previous run, r+1 is the next run

S^{min} = minimum storage that occurred (i.e., storage at minimum level difference date)

$S^{BottomConPool}$ = storage corresponding to an empty conservation pool

A = average yield distribution factor over the drawdown period. The yield may be distributed to the Reservoir Diversion based on factors representing the percentage/fraction of the average yield. These factors are stored in the Distribution Yield periodic slot and must average to 1.0 over the course of a year. Otherwise an error is issued and the run stops. Because of this distribution, the average of the yield over the *critical period* may not be 1.0. In this equation, we divide by the average distribution factor over this period to ensure that we are still dealing with average yield.

Δt = duration of the drawdown period, t_{begin} to t_{end}

E = total evaporation over drawdown period

$\hat{E}$ = estimated total evaporation that would occur over the drawdown period if the true yield were released. Increasing the yield from the last run will lead to lower storage values over the drawdown period, which will in turn lead to a lower evaporation over the drawdown period. We estimate $\hat{E}$ by computing the evaporation which would result from a drawdown period whose Storage ($\hat{S}$) is like the previous run's Storage except that it is scaled so that it reaches the bottom of the conservation pool at the end of the drawdown period (i.e., the blue dotted line in the figure). The estimate for the storage during the drawdown period when we are using the true yield is:

$$\hat{S}(t) = \frac{S_r(t)}{S_r^{BottomConPool}(t)} \cdot \frac{t - t_{begin}}{t_{end} - t_{begin}} \quad \textbf{(EQ 7)}$$

where:

t_{begin} = the begin date of the drawdown period

t_{end} = the end date of the drawdown period

And then, we can use this along with the method for computing evaporation as a function of storage, Evap(S(t)), to compute the total estimated evaporation over the drawdown period:

$$\hat{E} = \sum_{t_{begin}}^{t_{end}} Evap(\hat{S}(t)) \quad \textbf{(EQ 8)}$$

7.3.3 Heuristic B

In Heuristic B, the minimum and maximum yields are computed to bound the solution, but no runs are made with the values. Instead the average of them is used on the first run (If the **Reservoir Data for Bisection.Initial Yield to Try** is not specified). Then after each successful run, regardless of whether the minimum level difference is positive or negative, the y_{r+1} is computed as above (in Equation 6) and used on the next run. On unsuccessful runs, a bisection is used.

7.3.4 Which method to use?

The following table summarizes the strengths and weaknesses of each algorithm. This is particularly important for long model runs where any extra runs means hours of computations. own.

ALGORITHM	STRENGTHS	POTENTIAL WEAKNESS
Bisection	Most robust and easy to understand. Requires the least amount of effort if evaporation or other physical processes are involved.	Potentially slow as runs are made to ensure the upper and lower bounds encompass the solution point. Also slower as it just keeps going until a solution is found. No information from the run is used to guide the next run.
Heuristic A	Potentially faster than bisection. Like Heuristic B, information from this run can be used to guide the trial in the next run.	Because the heuristic is only used for positive minimum elevation differences, the search may be close to the final solution but then has a negative minimum elevation, so then uses bisection and jumps away from the final solution.
Heuristic B	Faster; the first two runs for each reservoir are skipped and the estimated yield is always used for the next run.	Because the upper and lower bounds are not tried first, the solution could be outside the bounds and would not be caught until hitting max iterations and/or attempting to converge on the bound.

8. Performance

Run time of the USACE-SWD models is always an important feature to consider. General tips and approaches can be found **HERE (Performance.pdf, Section 1)**.

8.1 User Tips

Following are specific tips for USACE-SWD models:

- Make sure each rule is not executing more often than necessary. Typically, each rule only needs to execute once per timestep. Use the approach documented **HERE (USACE_SWD.pdf, Section 3.2.4)** to control this.
- When diverting from reservoirs using the ComputeReservoirDiversions function, make sure that Reservoir.Available For Diversion is NOT linked to Diversion.Available For Diversion. This link is unnecessary and can lead to excessive dispatching of the diversion object. Click **HERE (USACE_SWD.pdf, Section 3.8.1.5)** for more information on the linking structure.
- If not debugging, disable the following as they are computationally intensive and are typically not necessary:
  - Diagnostics
  - RPL Set Analysis
- Close any unused open windows, particularly the Model Run Analysis tool. This and other dialogs attempt to redraw after every timestep and can slow down the model.
- Close other applications that are competing for computer resources.
- Consider using 3Gig tuning **HERE (Performance.pdf, Section 5.3)**

When the Diversions from Reservoirs conservation operation was initially implemented, the reservoir and the diversion object required the following links:

Reservoir.Diversion <---> Diversion.Diversion **(EQ 9)**

Reservoir.Available For Diversion <---> Diversion.Available For Diversion **(EQ 10)**

This was required because the diversion object had Available For Diversion as one of the required knowns in its dispatch conditions.

It was later found that the link between the Available For Diversion slots was leading to issues when the reservoir was low. To fix this, we removed Available For Diversion as a required known in the Diversion object's dispatch method. This was implemented in RiverWare 4.9. With this change, the link between the slots is no longer necessary.

The current analysis shows that there is unnecessary dispatching of the diversion objects because of this link. Any time the reservoir sets a new storage, the Available For Diversion at the next timestep is set. With the link in place, the value propagates to the Diversion object and the Diversion object redispatches, which then can cause the reservoir to redispatch. Thus, the Diversion objects are dispatching over the forecast period unnecessarily. Remove this link if they exist.

8.2 Internal RiverWare changes

The following sections describe changes to the code to specifically improve performance in USACE-SWD models.

8.2.1 No Dispatching Beyond Forecast Period

The default behavior in RiverWare is for objects to dispatch whenever they have adequate information to do so. It was noted in the analyses that objects were dispatching well into the future beyond the end of the forecast period at the current timestep due to the step response routing method on reaches that distributes a current inflow into outflows at future timesteps. No information beyond the forecast period is used for calculations at the current timestep, and information from dispatching beyond the forecast period is overwritten anyway by subsequent calculations as the model steps forward. Dispatching beyond the end of the forecast period is, therefore, not necessary.

To limit dispatching to the forecast period only, changes were made to the controller. At the beginning of the run, the flood basin computational object tells the controller what the forecast period is. If there are multiple flood basin computational objects, the controller uses the largest forecast period. At the beginning of each timestep, the controller calculates the end time of the forecast period by adding the forecast period to the current controller time. A condition was added to the checkDispatching method so that if the date of the dispatch is beyond the end time of the forecast period, the dispatch entry is not put on the dispatch queue for solving.

8.2.2 No Setting of Values Beyond Forecast Period in Step Response Routing

The step response routing method in reach calculates and sets outflow values into future timesteps. If the timesteps are beyond the end of the forecast period, this work is unnecessary because this information is not needed in the model for solving at the current timestep.

The step response routing method was modified so that if forecasting is being used, the reach asks the controller for the end time of the forecast period. The step response calculations are then limited so that outflow values are not calculated or set beyond the end of the forecast period.

8.2.3 Cache Routing Vectors

On the computational subbasin, there was significant time spent getting the appropriate control point and looking up the routing vector on its routing coefficients slot for the upstream reservoir.

Instead of doing this lookup each time, the code was changed to cache the routing vector information on a data structure in the computational object the first time coefficients are requested for a particular upstream and downstream object pair. Subsequent requests can then access the routing information from the data structure instead of redoing the lookup. If variable routing coefficients that can change with each timestep are being used, the data structure is cleared at the beginning of each timestep. Otherwise, the data structure can be used throughout the whole run.

8.2.4 Custom Downstream Dispatch Order

In analyzing dispatching, it was noted that excess dispatching occurs where values have been set on a number of different objects across the model. Dispatching order in RiverWare is based on a queue where objects dispatch in the order in which they come onto the queue. For example, in flood control where a value is set on each reservoir in the system, the reservoirs all go onto the queue and each reservoir dispatches in order before any objects dispatch that have come onto the queue as a result of the reservoir dispatches. This means the effect of an upstream reservoir's dispatching is not propagated to the downstream reservoir before it dispatches the first time, so it must go back on the queue to dispatch again (and maybe a third or fourth time due to additional upstream reservoirs).

To address this problem, a custom downstream dispatch order was implemented. The first time custom dispatching is called, the rule controller creates an ordered upstream to downstream list of objects in the network based on the topology information of upstream and downstream links between objects. Custom dispatching then iterates the dispatch queue successively for each object in the ordered list and dispatches all of that object's entries so that a downstream object does not dispatch before upstream effects have been propagated to it.

There is overhead associated with iterating the queue by object, so custom dispatching is only called in places where a number of objects across the basin are being put on the queue at the same time. For the COE model this has been implemented for computing incremental local inflows, the flood control rule, and the hydropower rule.

Note that the flood control algorithm is sensitive to small changes in input values, even changes that may be within convergence of slots. Changing the dispatch order of objects can affect how very small changes get propagated downstream or not and can change results by kicking off an operation at an earlier or later timestep. The custom dispatch order actually ensures that even small changes are propagated upstream to downstream, and due to this, gives different results at particular timesteps compared to the baseline executable. However, even though results at individual timesteps may differ, the overall trends and form of the solution is the same.