RiverWare

Technical Documentation Version 7.3

Accounting

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

Table of Contents

Chapter 1
Accounting Overview

This chapter provides detailed information on the Accounting system and how it works including background material, definitions, solution algorithms, and description of accounting methods and slots. In particular, first we describe the requirements for water accounting. Then we present the different types of accounts, their slots, solution algorithms and properties of accounts. We then discuss supplies, object level accounting methods, and the controllers used to solve the accounting system. Next, we describe how to use the accounting user interface and some of the utility features. We describe exchanges and give some examples and finally we present the water rights allocation solver.

This chapter covers the following topics:

Motivation and Requirements

River and reservoir basins are operated for various purposes including power production, irrigation, environmental purposes, recreation, and water quality. In many of those basins, it is necessary for water managers to track not only the volume and flow of water throughout the basin, but managers must also track the water ownership and type of the water. Operating decisions in the basin are dependent on many aspects including a user's available water, legal restrictions, physical constraints, and any exchange mechanism. In addition, many of the basins in the United States are governed by the doctrine of Prior Appropriation which says that the right to use a water supply is based on "First in time, first in right." Because of these constraints, it is necessary for water managers to have a tool to simulate the operating decisions and their effect in the basin including water type and water ownership.

The requirements for such a simulation tool are as follows:

- Must track ownership and type of water through all basin features and at all timesteps
- Flexible to model accounting in any basin with unique policies and structure
- Operating decisions must look at and set account, transfer and exchange values
- Must be able to allocate water based on the Water Right Priority date
- Must be able to visualize the accounting network

Water Accounting in RiverWare

In RiverWare, the above requirements are met by water accounting. Simply, water accounting is a layer added to a simulation model used to track the ownership and type of the simulated water. The following box highlights how the above requirements are met by RiverWare's water accounting functionality. These pieces are described in the following text as an introduction to water accounting and in more detail in this document.

Water accounting in RiverWare

- Physical and paper water are modeled separately.
- There is a separate network of accounts on the simulation objects.
- Accounts are linked indicating the possible transfers.
- Legal Accounts – Storage, Diversion, Instream Flow.
- Non Legal - Passthrough accounts track transfer of water between legal accounts.
- Accounts are labeled by ownership and type and can be given a priority date.
- Rules can access accounting information and also set account transfers.
- Can simulate water accounting components like accrual, exchanges, carryover, allocation, etc.
- A solver allows the user to allocate water to legal accounts based on the priority date.

In an accounting model, accounts, slots, and data are added to track why water is released, stored, or diverted. These additional components include water ownership, type, and purpose as water moves through the basin. This network is separate from the simulation network but there are methods that allocate physical water to the accounting network.

Accounts are linked to one another on both the same object and on different objects. These links indicate the possible transfers of water in the system. By defining a link, the user indicates that at some timestep, water could move between the two linked accounts. Legal accounts, including Storage, Diversion, and Instream Flow, are used to model a legal water right. Passthrough accounts are used to track the transfer of water through basin features. With both legal and non-legal accounts, the user can look at any object in the basin and determine the type and ownership of all of the water in that object.

Rules can access the account solution but can also control and set the releases from the accounts. User inputs can also drive the account solution. Using rules and other accounting utilities, the user can simulate typical accounting functionality including accrual, exchanges, carryover, and allocation. In addition, a special predefined rule function and set of methods serves as a water rights allocation solver to allocate water to legal accounts based on prioritized water rights.

Model Purpose

Water accounting can be used for a number of purposes. These include after-the-fact accounting, daily operations, and long term planning as described as follows:

After-the-fact Accounting

An after-the-fact accounting model is used on a timestep by timestep basis to track the type and ownership of water that was released the preceding timestep. For example, a water district might release water for a number of specific purposes on a given day. The following day the gaged data is imported into the model and final storages, evaporations, and losses can be calculated. The water district is able to then track how those actual releases and losses in the system should be charged to the various uses. This is considered an after-the-fact accounting model because the physical volumes, losses, and flows are known; the accounting model must calculate how those operations should be charged to the users in the basin. Often this is an exercise in book-keeping.

Short-term Operations

A short-term operations model uses information from the previous days and forecasts to determine the operations for future timesteps. If the operations depend on the amount of water available to specific users in the basin, then an accounting system is necessary to track how those releases will be made. In practice, rules set releases after looking at the balance on a certain account. For example, if a user requests a release, but that user has no available water in the accounting system, no physical release can be made.

Long-term planning

Long term planning models are used to determine the effect operations have on a longer time scale. If the operations depend on water ownership or type in the basin, then an accounting model can be used to track those pieces. The model must look at the state of the system including water ownership and type at each timestep and determine the operations to perform.

Often, water managers may use each of the three types of accounting models in a basin for water management. An after-the-fact accounting model is used each timestep to account for the previous timestep. The results from the after-the-fact model are fed into a short-term operations model to determine upcoming operations. Finally, a long term planning model is used for impact analysis, yield determination, and other planning related purposes. The structure, i.e. the physical layout of objects and accounts in the model, are the same in each model, only the operating policy rules and data are different.

Physical versus Paper Water

Let us develop some definitions to better discuss the accounting system. *Physical Water* is modeled using the simulation objects and does not have a unique classification. For example, the outflow from a reservoir is considered physical water. It may be released for a number of downstream purposes, but the total outflow is the value on the Outflow slot.

> Physical Water
>
> Water that is simulated by the RiverWare objects. This water represents the total Inflow, Outflow, and/or Storage not considering the use, ownership, or type of the water.

To contrast physical water, in the accounting system there is *Paper Water* (sometimes called colored water). This is the water that has a specific owner or type that must be tracked.

> Paper Water
>
> Paper water is simulated in the accounting system to track ownership or type. It is called paper water because it is often tracked on paper and may be one of many components of the physical water.

Accounts

To track the paper water that is moving through the system, accounts are created on the simulation objects. An account is an object used to track the inflow, outflow, and storage of paper water on the simulation object. Like simulation objects, given the required knowns, the accounts can solve for certain unknowns.

> Account
>
> An account is an object used to track the balance of paper water with a given type and/or ownership on a simulation object.

Accounts can be defined as either a legal account or a passthrough account:

- Legal Accounts: There are three types of legal accounts, Storage, Diversion, and Instream Flow. These accounts can have a legal right to store water, divert/consume water or have a certain flow rate of water in the river, respectively.
- Passthrough Accounts: Passthrough accounts are used to connect legal accounts and are used to show how much paper water is on an object at a given timestep. They do not have a legal right.

The types of allowable accounts are dependent on the type of simulation object on which it resides.The four types of accounts, Storage, Diversion, Instream Flow and Passthrough accounts, are described in the following sections. For each account type, the following is described:

Slot Descriptions

Each account has general and/or method specific slots. Following is a key for the description:

+ **Slot Name**

Type: The type of the slot, including Series, Agg Series, Multi Slot, Scalar, Table, List, or Periodic

Units: The primary unit type. This is the unit type used to store the value internally. If setting the slots with rules or Object Level Accounting Methods, this unit type must be used.

Alt Units: The alternative unit type used for display purposes. For example, if Flow is the primary unit type, then Volume (over the timestep) is the alternative unit.

Description: A description of the slot.

I/O: Input or Output type. See the table below for the types.

Supply Links: Yes or No. Can this slot be linked to another slot in the accounting system using a supply.

I/O Types	Description or Example	Flag
Input	A value set directly by the user via the user interface or DMI’s. If the slot is linked via a supply, the supplies should be set as user input and will propagate to the connected slots.	I

I/O Types	Description or Example	Flag
Account Level Method	The value is set by an Account Level Method. For example, Carryover is set based on the selected Storage Account Carryover method.	m
Account Solution	The value is set by solution of the account. For example, outflow to a passthrough account on a reach is calculated as the sum of the inflow, slot inflow, gain loss and return flow minus the diversion.	A
Propagated	The value is set through propagation of a value across a supply link. The other end of the supply is calculated by a method or the slot is linked to multiple supplies where one or more is set by a rule or input. The summation column of a multislot will have P flag even though individual columns may have an I or R flag.	P
Object Level Accounting Method	The value is set by an Object Level Accounting Method, either compiled or user defined.	m
Rule	A value may be set by a rule only if there is no supply connecting the slot to another slot. If there is one or more supplies connecting the slot, then the rule must set the supply(ies). The value and flag is reflected in the linked slots' values as either an R flag (one supply) or a P flag (two or more supplies connected via the multi-slot).	R

User method categories

Accounts can have user selectable methods that control how the accounts behave. For example, the user can select a carryover method where all of the storage in the account carries over from one water year to the next. Alternatively, the user can configure the account such that the storage in the account is reset at the beginning of the water year. Provided for each account is a description of each category and the methods in that category including the method specific slots.

Account Solution Equations

Accounts solve when they have the required information. Unlike simulation, accounts always solve in the "downstream" direction. Storage accounts always solve for Storage, passthrough accounts usually solve for Outflow, but never for Inflow. Each type of account is described in more detail in the following sections including the solution equations for each type of account. Additional information on how the accounting system solves is provided in "How the Accounting System Solves" on page 76.

Storage Account

A Storage Account represents a legal storage right in a reservoir. It can be used to track the accumulation and release of water that represents that right. Storage accounts are allowed on any of the reservoir objects, including Storage Reservoirs, Level Power Reservoirs, Slope Power Reservoirs, and Pumped Storage Reservoirs.

Storage Account Slot Descriptions

Following is a description of the general slots that reside on a Storage Account. Additional slots may be instantiated depending on user methods. Slot characteristics (Type, Units,...) are described in “Slot Descriptions” on page 13.

+ Accrual

Type:	Series Slot
Units:	Volume
Alt Units:	NA
Description:	Cumulative volume of inflows to the account from the Begin Accrual Date to the current timestep
I/O:	Input (at initial timestep) or Account Solution
Supply Links:	No

+ Begin Year Allocation

Type:	Series Slot
Units:	Volume
Alt Units:	NA
Description:	Volume allocated to account on begin Accrual Date. This value is then used in the solution equation in place of the previous storage. The begin year allocation allows the user to specify the initial storage for the accrual period.
I/O:	Input or Rules
Supply Links:	No

+ Carry Over

Type:	Series Slot
Units:	Volume
Alt Units:	NA
Description:	Volume of water carried over to the Begin Accrual Date from previous storage
I/O:	Input, Rule, or set by the Account Level Method
Supply Links:	No

+ Diversion

Type:	Multi Slot
Units:	Flow
Alt Units:	Volume
Description:	Water diverted out of the storage account and transferred to a diversion account
I/O:	Input or set by a rule
Supply Links:	Yes, to Diversion slot on a Diversion account

Gain Loss

Type:	Series Slot
Units:	Volume
Alt Units:	Flow
Description:	A loss or gain due to local effects including evaporation, seepage, precipitation. This gain or loss represents this account's allocation of the physical loss on the reservoir object. A gain is positive, a loss is negative.
I/O:	Input, Rule, or Object Level Accounting Method
Supply Links:	No

Inflow

Type:	Multi Slot
Units:	Flow
Alt Units:	Volume
Description:	Water transferred into this account from upstream
I/O:	Propagated from upstream
Supply Links:	Yes, from an upstream account

Maximum Accrual[1]

Type:	Series Slot
Units:	Volume
Alt Units:	NA
Description:	This slot represents the maximum accrual that is allowed during the accrual period. Maximum Accrual is not used in the account solution or in the water rights allocation but is available to hold data for viewing and use in Object Level Accounting Methods and Rules.
I/O:	Input or Rule
Supply Links:	No

Outflow

Type:	Multi Slot
Units:	Flow
Alt Units:	Volume
Description:	Water transferred out of this account downstream
I/O:	Input or Rule
Supply Links:	Yes, to a downstream account

Return Flow

Type:	Multi Slot

1. Maximum Accrual is not yet used by SolveWaterRights()

Units:	Flow
Alt Units:	Volume
Description:	Water entering the storage account as return flow from a diversion account
I/O:	Input, Rule, or Propagated
Supply Links:	Yes, to Return Flow slot on a diversion account

+ Slot Inflow

Type:	Series Slot
Units:	Flow
Alt Units:	Volume
Description:	Water added to the account from a local source. This Slot Inflow represents this accounts allocation of the local inflows that enter the reservoir object
I/O:	Input, Rule, or Object Level Accounting Method
Supply Links:	No

+ Storage

Type:	Series Slot
Units:	Volume
Alt Units:	NA
Description:	Storage balance in the account
I/O:	Input (at initial timestep) or Account Solution
Supply Links:	No

+ Transfers In

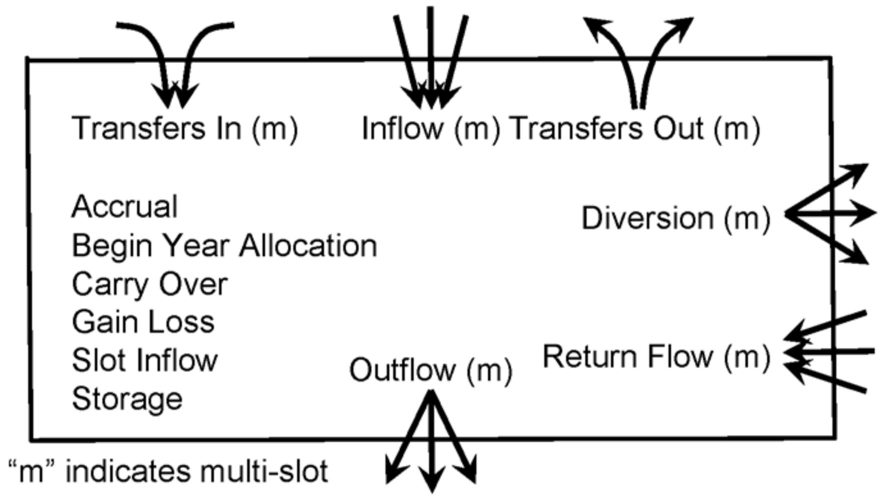
Type:	Multi Slot
Units:	Flow
Alt Units:	Volume
Description:	Water transferred into this account from another account on the same object
I/O:	Input, Rule, or Propagated
Supply Links:	Yes, from a Transfers Out slot on another account on the same object

+ Transfers Out

Type:	Multi Slot
Units:	Flow
Alt Units:	Volume
Description:	Water transferred out of this account to another account on the same object
I/O:	Input, Rule, or Propagated
Supply Links:	Yes, to a Transfers In slot on another account on the same object

Figure1-1 is a diagram that shows the general slots on the Storage Account. The arrows leading into and out of the account represent supplies. The “m” indicates that the slot is a multi-slot and can be linked to more than one other slot and hence, there are multiple arrows.

Figure 1-1: General slots on the Storage Accounts



Storage Account User Method Categories

Following are the user method categories on the Storage Account.

- "Initial Request."
- "Conservation Pool Fill Factor" on page 21.
- "Appropriation Request Adjustment" on page 21.
- "Storage Account Carryover" on page 22.
- "Subordination of Right" on page 22.
- "Water Right" on page 22.
- "Min Bypass" on page 24.

Initial Request

This category is used to specify how the storage account calculates the Initial Request in the water rights allocation. The methods in this account are only valid if the account has a water right as specified by the Priority Right method.

Fill Conservation Pool

This method allows the user to specify that the appropriation of water to this account is the amount of water that it takes to fill the conservation pool. The total volume of paper water in all storage accounts on a reservoir contributes to the conservation pool volume, regardless of the ownership of the water in these accounts.

The amount of water required to fill the conservation pool is calculated as follows: At the beginning of each timestep, the reservoir object determines a value to place in its series slot Conservation Pool Initial Empty Space if that slot does not already have a value for the current timestep. Note, the reservoir must have one of the "Conservation..." methods selected in the Operating Levels category. The value calculated is the amount of inflow needed to fill the conservation pool. This flow is computed by calling the reservoir's utility function massBalanceSolveInflow() given zero outflow and the elevation of the top of the conservation pool, as defined by the slots Top of Conservation Pool, Operating Level Table, and Elevation Volume Table. The mass balance utility method takes into account all the method selections on the reservoir, including those for evaporation and precipitation, and the current Pool Elevation of the reservoir. The value in the Conservation Pool Initial Empty Space is based on data on the simulation object (i.e. "wet" water) and does not take into consideration any data on the accounts ("paper" water).

When the water allocation solver computes the physical constraints applicable to a storage right and when it computes the appropriation request for a storage right that has the Fill Conservation Pool method selected, it starts with the current value of the Conservation Pool Initial Empty Space slot on the underlying Reservoir object. If this value is not defined, and the solver in use is SolveWaterRightsWithLags, the solver will force the reservoir to compute this value. From that value it subtracts inflows, slot inflows, and net transfers in (subtracts transfers in, adds transfers out) to all the storage accounts on the reservoir; the remainder is the inflow that will fill the conservation pool on the reservoir. If this value is less than zero, zero is used. This is the desired initial appropriation request. This initial appropriation request amount is subject to the optional Fill Factor (q.v., below).

When the Fill Conservation Pool method is selected, the following slots are added:

Initial Request

Type:	Series Slot
Units:	Flow
Alt Units:	Volume
Description:	The water rights initial appropriation request. This is how much water the right wishes to take from allocatable flow with legal and physical constraints applied. This slot is given the value of the Appropriation Request slot only when this slot's value is not defined. Thus, it is meant to be a saved copy of the Appropriation Request slot from the first call to the solver -- note that this use of the Initial Request slot is not consistent with its use on other account types, nor is it consistent with the slot's use when the Specify Initial Request method is selected.
I/O:	Input or Rules
Supply Links:	No

Fill Conservation Pool with Diversions

This method is similar to the Fill Conservation Pool method (above), but it allows more water to be requested in order to meet the Initial Requests of Diversion Accounts on Water Users that are directly supplied by this account and its sibling storage accounts.

The amount of water required to fill the conservation pool is calculated exactly as for the Fill Conservation Pool method. To this amount is added the sum of all the Initial Request values on all Diversion Accounts supplied by this storage account or by its sibling storage accounts. If this amount is less than zero, zero is used.

From this value it subtracts inflows, slot inflows, and transfers in to all the storage accounts on the reservoir; the remainder is the inflow that will fill the conservation pool on the reservoir and meet diversions. Again, if this value is less than zero, zero is used. This is the desired initial appropriation request. This initial appropriation request amount is subject to the optional Fill Factor (q.v., below).

When the Fill Conservation Pool with Diversions method is selected, the following slots are added:

Initial Request

Type:	Series Slot
Units:	Flow
Alt Units:	Volume
Description:	The water rights initial appropriation request. This is how much water the right wishes to take from allocatable flow with legal and physical constraints applied. This slot is given the value of the Appropriation Request slot only when this slot's value is not defined. Thus, it is meant to be a saved copy of the Appropriation Request slot from the first call to the solver -- note that this use of the Initial Request slot is not consistent with its use on other account types, nor is it consistent with the slot's use when the Specify Initial Request method is selected.
I/O:	Input or Rules
Supply Links:	No

Specify Initial Request

In this method, the user can specify the Initial Request via Input, DMI, or Rules. If one of the non-default methods is selected in the Operating Levels category on the reservoir, the filling the conservation pool will also limit the allocation. This method is then available to further limit the allocation. When the Specify Initial Request method is selected, the following slots are added:

Initial Request

Type:	Series Slot
Units:	Flow
Alt Units:	Volume
Description:	The water rights initial appropriation request. This is how much water the right wishes to take from allocatable flow. The solver applies legal and physical constraints to this amount, to produce the Appropriation Request slot value.
I/O:	Input or Rules
Supply Links:	No

Shortage

Type:	Series Slot
Units:	Volume
Description:	The difference between the Initial Request and the transfers into the storage account.
I/O:	Calculated by an Account Level Method
Supply Links:	No

Conservation Pool Fill Factor

This category's methods allow a conservation pool to be filled in part (generally in order to allow other storage accounts on the object to receive some of the conservation pool paper-water). Storage Accounts with the Fill Conservation Pool method or the Fill Conservation Pool with Diversions method selected make use of the Fill Factor. In this category are the following methods:

None

No fill factor is given. 1.0 is used.

Fill Factor is Scalar

This method allows the user to specify a fixed fill factor in the scalar slot:

+ **Fill Factor**

Type:	Scalar Slot
Units:	No Units
Alt Units:	No Units
Description:	Fraction of the conservation pool (or conservation pool and diversion requests) to be requested in the Appropriation Request slot. If not defined, 1.0 is used.
I/O:	Input
Supply Links:	No

Fill Factor is Series

This method allows the user to specify a series of fill factors in the slot:

+ **Fill Factors**

Type:	Series Slot
Units:	No Units
Alt Units:	No Units
Description:	Fraction of the conservation pool (or conservation pool and diversion requests) to be requested in the Appropriation Request slot. If not defined, 1.0 is used.
I/O:	Input
Supply Links:	No

Appropriation Request Adjustment

This method is used to specify an adjustment to the appropriation request, which is computed by SolveWaterRights() from the Initial Request and from the state of the network.

Return Flow Credit

This method causes return flows to be subtracted from the computed appropriation request value before the latter is put into the Appropriation Request slot.

Storage Account Carryover

The methods in this category are executed on the Begin Accrual Date to specify how water is carried over from one year to the next. The Begin Accrual Date is specified in the Water Accounting System configuration.

No Carryover

Zero Storage is carried over from one year to the next.

Carryover All Storage

All Storage is carried over from one year to the next.

Pooled Carryover

This is a specific method for a reservoir with a storage account named “Pooled”. It carries over all of the water in the Pooled account but removes any Begin Year Allocation in other accounts on the object with the same water type.

Subordination of Right

This method allows an exception to the strict prior-appropriation priority determination to be respected by SolveWaterRights().

Subordinate Senior Rights

This method is used to specify that a downstream senior right may have to give up water in the event that this right would be shorted. More than one downstream senior right may be subordinated to this right, in which case, the subordinates give up water to satisfy this right in order of their relative priorities. This method is used to specify that a downstream senior right may have to give up water in the event that this right would be shorted.

+ Subordinated Rights

Type:	List slot of Accounts
Units:	No Units
Alt Units:	NA
Description:	This list slot holds the set of downstream senior rights that are subordinated to this right.
I/O:	Input
Supply Links:	NO

Water Right

Methods in this category specify whether the account has a water right.

None

No water right.

Priority Right

When selected, this account will have a water right. Selecting this method is identical to checking the “Has Priority Date” toggle in the account configuration, General tab. The following slot is added when this method is selected:

+ Appropriation Request

Type: Series slot
Units: Flow
Alt Units: Volume
Description: A result of the water rights allocation solver. This is the amount of allocatable flow that the water rights solver will try to appropriate to this account. This slot is given a value that is derived from the initial request, reduced by any applicable legal and physical constraints.
I/O: Rule
Supply Links: NO

Water rights may be subject to physical and legal constraints on their initial requests for water. The net requested water, after such constraints are applied, is put into the Appropriation Request slot by the water rights solver. Storage accounts are subject to one or two physical constraints: a) the appropriation may not cause the conservation pool to overflow, and 2) if the appropriation is also a diversion, the appropriation may not exceed the physical capacity of the diversion structure. The conservation pool constraint uses the Conservation Pool Initial Empty Space slot on the reservoir (see Fill Conservation Pool method in the Initial Request category for a description of how that slot is calculated). If the reservoir’s Conservation Pool Initial Empty Space slot is not available (because no method that uses it is selected), this physical constraint is not applied.

The water rights solver rule function, SolveWaterRights (or SolveWaterRightsWithLags) assigns a flow to a supply for a water right; this flow might be less than is requested in the Appropriation Request slot. The table below describes the various reasons the solver allocates what it does. These values may be OR-ed together.

+ Temp Reason

Type: Series slot
Units: None
Alt Units: NA
Description: This temporary slot encodes diagnostic information about the reason why the account is allocated the amount that it is given by the water rights solver.
I/O: Output only
Supply Links: No

I/O Types	Allocation is limited by	Description
0000000001	Initial Request, which was input.	
0000000002	Initial Request, which was to fill the conservation pool.	Conservation Pool Initial Empty Space minus the sum of all storages in sibling storage accounts (converted to flow) is the upper bound applied. Storage accounts only.

I/O Types	Allocation is limited by	Description
0000000010	Min Bypass criteria	Min Bypass criteria are reflected in the Appropriation Request.
0000000020	Appropriation Request adjustment was applied	Return flows reduce the Appropriation Request.
0000000040	Max Accrual	Not yet implemented.
0000000100	Conservation Pool capacity	May apply when Initial Request is Input

+ Temp Available For Shared Priority

Type: Series slot
Units: Flow
Alt Units: Volume
Description: A result of the water rights allocation solver. When the Share Proportionally with Limits method is selected in the Account Equal Priority Allocation method category on the computational subbasin, this slot is instantiated. If this account has a priority date equal to other accounts this slot is populated with the amount of water available to this account independently of the other shared priority accounts at the time water is allocated. This value is used in the calculations that ultimately allocate water to the shared accounts as described under the Water Rights Allocation Solver HERE (Section 10).
I/O: Rule
Supply Links: No

Min Bypass

This method is executed as part of a prioritized water rights allocation. This method models a legal constraint that may reduce a water right's initial request for water by forcing the water right to leave a certain amount of water in the stream at a given reference location.

None

No minimum bypass is defined for this account.

Fraction of Flow Above Min

The following slots are added for this method:

+ Bypass Reference Location

Type: List
Units: NA
Alt Units: NA
Description: Optional, control point at which you wish to keep a minimum bypass in the stream. If a control point is not specified in this slot, the accounting flow in the linked object will be used.
I/O: Input only, only one control point can be specified

Supply Links: No

Absolute Min Bypass

Type: Periodic
Units: TIME vs Flow
Alt Units: NA
Description: The required flow that must remain in the stream
I/O: Input only
Supply Links: No

Fraction above Min

Type: Periodic
Units: Time vs none
Alt Units: NA
Description: The fraction of the flow above the Absolute Min Bypass that must be left in the stream
I/O: Input only
Supply Links: No

The requirement contains two components, the Absolute Min Bypass and the Fraction above Min.

Following are three examples to show how this method calculates the flow:

- A legal requirement says to leave 300cfs in the stream. The following would be input: Absolute Min Bypass = 300cfs, Fraction above Min = 0.
- A legal requirement says to leave 45% of the flow in the stream. Input the following: Absolute Min Bypass = 0cfs, Fraction above Min = 0.45.
- A legal requirement says to leave 300cfs plus 45% of the flow above that minimum in the stream. Input the following: Absolute Min Bypass = 300cfs, Fraction above Min = 0.45.

This method is used to specify the flow to leave in the supply chain. This location can be specified in two ways depending on whether a control point is specified in the Bypass Reference Location:

- If the there is a control point specified in the Bypass Reference Location, the accounting flows on that object will be used.
- If there is no control point specified in the Bypass Reference Location, then the accounting flows on the object supplying the allocation will be used. That is, for an allocation to a storage account, the amount available for appropriation from the passthrough accounts on the reservoir will be used.

The required bypass flow is calculated using the:

- Sum of Temp Available for Appropriation slot for all passthrough accounts on the object (non-control points), or
- Sum of Outflow for all passthrough accounts at the control point reference location.

If the reference location is downstream, then any lags and/losses are applied so that the min bypass constraint includes enough water to meet the min bypass at the reference location. Then, the constraint (i.e. upper limit on the appropriation) is temporarily stored in the Temp Min Bypass Constraint slot on the diversion account but is not saved with the model file.

Storage Account Solution Equations

For each timestep in the accounting period where there are the required knowns, solve for Storage(t) when Storage(t-1) is known and time(t) is *not* the Begin Accrual Date.

Required Knowns: Storage(t-1), Accrual(t-1), Slot Inflow(t), Gain Loss(t),

$$Storage(t) = Storage(t-1) + GainLoss(t) + TimestepLength \times (Inflow(t) - Outflow(t) + SlotInflow(t) - Diversion(t) + ReturnFlow(t) + TransfersIn(t) - TransfersOut(t))$$

$$Accrual(t) = Accrual(t-1) + TimestepLength \times (Inflow(t) + SlotInflow(t))$$

If time(t) *is* the Begin Accrual Date, Solve for Storage(t):

Required Knowns: Storage(t-1), Accrual(t-1), Slot Inflow(t), and Gain Loss(t) if CarryoverAllStorage method is selected. Slot Inflow(t) and Gain Loss(t) if NoCarryover method is selected.

$$Storage(t) = BeginYearAllocation(t) + CarryOver(t) + GainLoss(t) + TimestepLength \times (Inflow(t) - Outflow(t) + SlotInflow(t) - Diversion(t) + ReturnFlow(t) + TransfersIn(t) - TransfersOut(t))$$

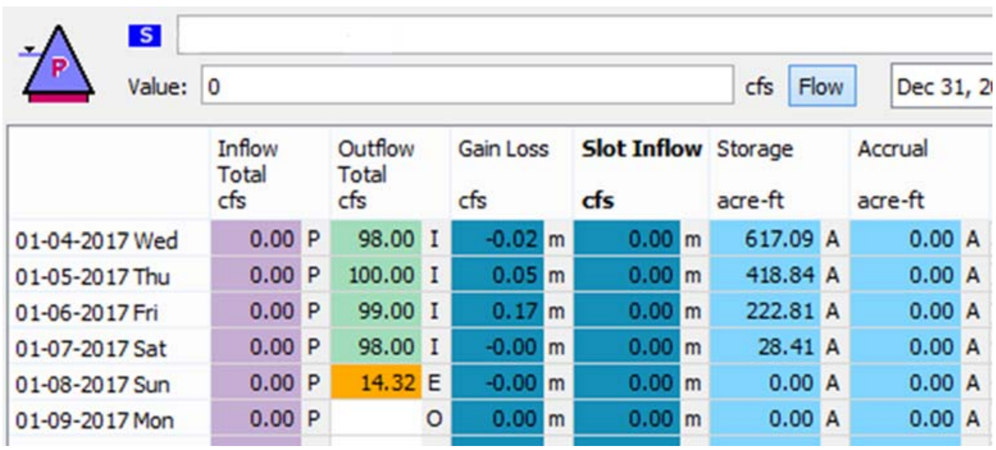
$$Accrual(t) = BeginYearAllocation(t) + TimestepLength \times (Inflow(t) + SlotInflow(t))$$

Note: *When the storage is solved in interactive mode, the account solved for as many timesteps as there are required knowns. During a run, the account solves for the current timestep and the following timestep. For models using the water rights solver, the storage account is allowed to solve through the current plus Local Timestep Offset, as defined on the sibling passthrough, to include the effects of lagging in the system. This difference was implemented for performance reasons.*

Empty Storage Flag

When setting supply values interactively, you may want to set a supply such that it completely empties a storage account. That is, you want all of the water in that Storage account to move downstream on a particular supply. Although you could type in the correct number or copy and paste the storage into the outflow supply (as a volume), this is error prone and would need to be repeated if other supplies change. Instead, use the “Empty Storage Flag”, denoted with the letter “E” to perform this operation. This flag computes the supply value that will lead to a zero storage.

Figure 1-2: Edit account dialog showing the Empty Storage Flag for a storage account with one outflow supply



	Inflow Total cfs		Outflow Total cfs		Gain Loss cfs		Slot Inflow cfs		Storage acre-ft		Accrual acre-ft	
01-04-2017 Wed	0.00	P	98.00	I	-0.02	m	0.00	m	617.09	A	0.00	A
01-05-2017 Thu	0.00	P	100.00	I	0.05	m	0.00	m	418.84	A	0.00	A
01-06-2017 Fri	0.00	P	99.00	I	0.17	m	0.00	m	222.81	A	0.00	A
01-07-2017 Sat	0.00	P	98.00	I	-0.00	m	0.00	m	28.41	A	0.00	A
01-08-2017 Sun	0.00	P	14.32	E	-0.00	m	0.00	m	0.00	A	0.00	A
01-09-2017 Mon	0.00	P		O	0.00	m	0.00	m	0.00	A	0.00	A

Following are some features of the E flag:

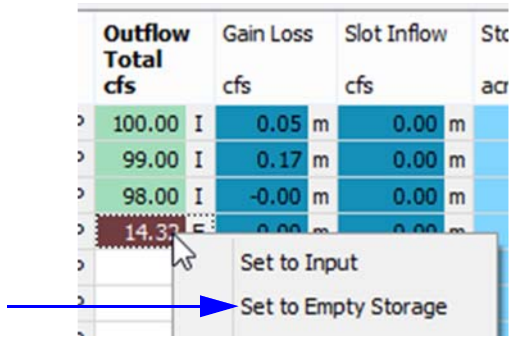
- The flag is associated with a timestep and a supply coming from a storage account. There are no restrictions on the downstream account type. Although it is possible to set the flag on other account’s supplies and accounting slots, it will affect a Storage account when set on exactly one outflow supply.
- When the flag is present on a supply, it computes and sets the supply value necessary to lead to a zero Storage at that timestep.
- The flag should be set on at most one outflow supply of a Storage account at each timestep. (Note setting it on more that one supply will lead to undesired behavior; the first supply set will take all of the storage).
- Once set, the E flag remains until explicitly cleared by the user.
- When a storage account unsolves, the E flag remains, the supply value is set to NaN.

Setting the flag

Interactively, the flag can be set on the supply in the following contexts:

- In the `Edit Account` dialog for the upstream storage account’s Outflow supplies, using either a right-click context menu or the `TimeStep I/O` > `Set to Empty Storage` menu option.

Figure 1-3: Right-Click context menu to set the Empty Storage Flag



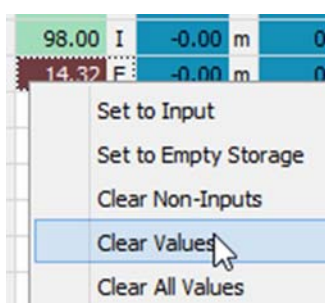
- In the open slot dialog for the upstream storage account's Outflow supplies, using either the right-click context menu or the `TimeStep I/O > Set to Empty Storage` menu option.
- In the account editor dialog for the downstream account, on the Inflow slot, if there is only one incoming supply for that account (and that supply comes from a Storage account).
- In the open slot dialog for the downstream account's Inflow slot, if there are more than one incoming supplies for that slot.
- In the open slot dialog for a supply, using the `TimeStep I/O > Set to Empty Storage` option.
- In the SCT dialog, when the outflow supply is shown, use the `Edit > Empty Storage` menu.

The flag cannot be set by a rule. In addition, the flag does not have a priority associated with it, so any rule can overwrite the flag and value.

Removing the flag

Once set, the E flag remains until explicitly cleared by the user. Remove the flag using the `Clear Values` operation.

Figure 1-4: Right-Click context menu to clear the Empty Storage Flag



Diversion Account

The diversion account represents a legal diversion right from a reach, distribution canal, or reservoir. Diversion Accounts can be created on Water User and Agg Diversion Site objects. On Agg Diversion Sites, Diversion Accounts are allowed on the aggregate or the water user elements based on the linking structure (i.e. the `Link Structure` menu). The following are the limitations:

- `No Structure`: Diversion Accounts are allowed on the Water User elements, but not the aggregate.
- `Sequential Structure`: Diversion Accounts are allowed on EITHER the Water User elements or the aggregate, but not both.
- `Lumped Structure`: Diversion Accounts are allowed on the aggregate, but not the Water User elements.

These limitations are enforced when creating accounts, but you can switch linking structures after creating the accounts. A warning message is displayed when you switch structures and have invalid accounts. The invalid accounts remain but are not saved in the model. During the saving process, a red message warns of any invalid accounts that have not been saved.

Diversion Account Slot Descriptions

Slot characteristics (Type, Units,...) are described in “Slot Descriptions” on page 13.

+ Accrual

Type: Series Slot
Units: Volume
Alt Units: NA
Description: Cumulative volume of water depleted from the Begin Accrual Date to the given timestep
I/O: Account Solution
Supply Links: No

+ Depletion

Type: Series Slot
Units: Flow
Alt Units: Volume
Description: Water consumed by the given account
I/O: Input, Rules, Object Level Accounting Method, or Account Solution
Supply Links: No

+ Diversion

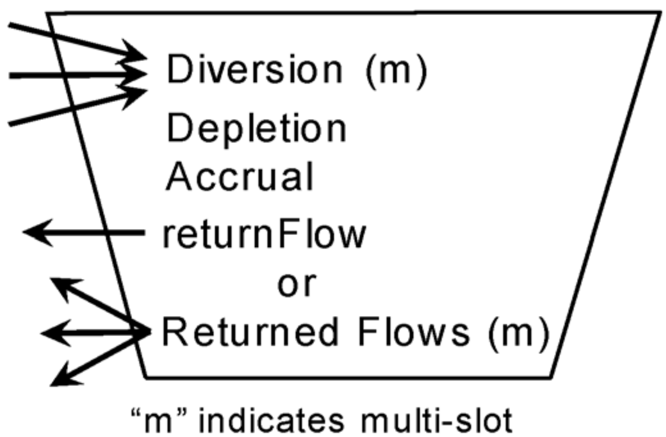
Type: Multi Slot
Units: Flow
Alt Units: Volume
Description: Water Diverted from a passthrough or storage account on another object.
I/O: Input, Rules, Object Level Accounting Methods
Supply Links: Yes, from Diversion slot on another account

+ Maximum Accrual[1]

Type:	Series Slot
Units:	Volume
Alt Units:	NA
Description:	This slot represents the maximum accrual that is allowed during the accrual period. Maximum Accrual is not used in the account solution or in the water rights allocation but is available to hold data for viewing and use in Object Level Accounting Methods and Rules.
I/O:	Input or Rules
Supply Links:	No

Figure1-5 is a diagram showing the general slots on the Diversion Account. Note, returnFlow and Returned Flows are not general slots, but are based on the selected Route and Split method. They are shown here because they can be linked. The arrows leading into and out of the account represent supplies. The "m" indicate that the slot is a multi-slot and can be linked to more than one other slot and hence, there are multiple arrows.

Figure 1-5: General slots on Diversion Account



Diversion Account User Method Categories

Following are the user method categories on the Storage Account.

- "Initial Request."
- "Return Flow Calculation" on page 32.
- "Return Flow Route or Split" on page 36.
- "Subordination of Right" on page 39.
- "Water Right" on page 39.

1. Maximum Accrual is not yet used by SolveWaterRights().

- "Min Bypass" on page 41.
- "Max Legal Request" on page 43.

Initial Request

This method is used to specify how the diversion account calculates the Initial Request in the water rights allocation. The methods in this category are only valid if the account has a water right as specified by the Priority Right method.

None

No initial request is necessary or specified. This is the default.

Specify Initial Request

When the Specify Initial Request method is selected, the following slots are added:

+ Initial Request

Type:	Series Slot
Units:	Flow
Alt Units:	Volume
Description:	The initial water rights appropriation request. This amount is used by the water rights solver to compute the Appropriation Request slot value. The Initial Request is the amount of water this right would like to appropriate, without consideration of physical and legal constraints such as the capacity of the diversion structure, or accrual maxima.
I/O:	Input or Rules
Supply Links:	No

+ Shortage

Type:	Series Slot
Units:	Volume
Description:	The difference between the Initial Request and the diversions coming into the account.
I/O:	Account Method
Supply Links:	No

This method allows the user to specify the initial appropriation request for use in a prioritized water rights allocation.

Disaggregated by Subbasin

The initial request is calculated from annual values. The following slots are added for this method:

+ Initial Request

Type:	Series Slot

Units: Flow
Alt Units: Volume
Description: The initial water rights appropriation request
I/O: Input or Rules
Supply Links: No

+ Shortage

Type: Series Slot
Units: Volume
Alt Units: Flow
Description: The difference between the Initial Request and the diversions coming into the account.
I/O: Account Method
Supply Links: No

+ Annual Request

Type: Series Slot, Annual timestep
Units: Flow
Alt Units: Volume
Description: The aggregated annual request values from which the Initial Request values will be derived by a computational subbasin at the beginning of a run.
I/O: Input. Required if this method is selected.
Supply Links: No

Annual requests are disaggregated to this model's timestep according to the values specified in the computational subbasin. This method is executed from the computational subbasin at the beginning of the run. For more information, see "Account Initial Request" in Objects.pdf.

Max Permitted

The initial request is assumed to be unbounded, so that the resulting Appropriation Request slot value is the maximum amount permitted by all other constraints, such as the amount of water available in the supply chain, physical constraints such as diversion structure capacity, and legal constraints such as minimum bypass, *and not shorting downstream senior accounts*, *unlike the calculations of Appropriation Request with all other method selections*. The Initial Request and Shortage slots are not made available when this method is selected. When this method is selected, the permitted appropriation is left in the Appropriation Request slot, but *no appropriation is made*.

Return Flow Calculation

The Return Flow Calculation category allows the user to specify how the return flow to the account is specified or computed. It also indicates how the account should solve based on the knowns and unknowns. In each case, the methods set the temporary variable tempReturnFlow[t]. This variable is then split and/or routed according to the method selected in "Return Flow Route or Split" on page 36. Below is a description of each method and the slots added by that method.

Diversion Minus Depletion

This method assumes that either Diversion and Depletion or Diversion and returnFlow are known. Then, the other is solved for.

If Depletion is known, compute the temporary variable tempReturnFlow[t]:

$$tempReturnFlow[t] = (Diversion[t] - Depletion[t])$$

If returnFlow[t+lag] is known, solve for tempReturnFlow[t] (this is only valid if using the Simple Lag method; if not an error will be issued).

$$tempReturnFlow[t] = \frac{returnFlow[t + lag]}{TimestepLength[t]} \times TimestepLength[t + lag]$$

If neither returnFlow[t+lag] nor Depletion[t] is known, assume that diversion will equal depletion, and compute the temporary variable tempReturnFlow[t]:

$$tempReturnFlow[t] = 0.0$$

Then, the tempReturnFlow[t] is split and/or routed according to the selected method in the Return Flow Route or Split method.

Note: *For more information, see* *"Return Flow Route or Split" on page 36.*

Specify Return Flow

This method allows the user to input the returnFlow. This method is only valid if using the Simple Lag method; if not an error will be issued.

If returnFlow[t+lag] is known, compute the temporary variable tempReturnFlow[t]

$$tempReturnFlow[t] = \frac{returnFlow[t + lag]}{TimestepLength[t]} \times TimestepLength[t + lag]$$

If returnFlow[t+lag] is unknown, compute the temporary variable tempReturnFlow[t]

$$tempReturnFlow[t] = 0.0$$

Then, the tempReturnFlow[t] is split and/or routed according to the selected method in the Return Flow Route or Split method.

Note: *For more information, see* *"Return Flow Route or Split" on page 36.*

Fractional Return Flow

This method allows the user to input a constant fraction of the diversion that is returned.

+ Fraction of Diversion

Type:	Scalar Slot
Units:	None
Alt Units:	NA
Description:	Fraction of the Diversion that is returned
I/O:	Input Only
Supply Links:	No

This method computes the temporary variable tempReturnFlow[t]:

$$tempReturnFlow[t] = FractionOfDiversion \times Diversion[t]$$

Then, the tempReturnFlow[t] is split and/or routed according to the selected method in the Return Flow Route or Split method.

Note: *For more information, see* *"Return Flow Route or Split" on page 36* *for more information.*

Variable Fractional Return Flow

This method allows the user to input a series slot that contains the fraction of diversion that is returned.

+ Variable Fraction of Diversion

Type:	Series Slot
Units:	None
Alt Units:	NA
Description:	Fraction of the Diversion that is returned
I/O:	Input Only
Supply Links:	No

This method computes the temporary variable tempReturnFlow[t]:

$$tempReturnFlow[t] = VariableFractionOfDiversion(t) \times Diversion[t]$$

Then, the tempReturnFlow[t] is split and/or routed according to the selected method in the Return Flow Route or Split method.

Note: *For more information, see* *"Return Flow Route or Split" on page 36* *for more information.*

Variable Efficiency Return Flow

This method allows the user to compute the return flow as a function of the specified Maximum Efficiency, Depletion Requested and Diversion

+ Depletion Requested

Type: Series Slot
Units: Flow
Alt Units: Volume
Description: Desired amount of water to be consumed by the diversion account

I/O: Input or Rules
Supply Links: No

+ Efficiency

Type: Series Slot
Units: Fraction
Description: The computed efficiency.

I/O: Output only
Supply Links: N/A

+ Maximum Efficiency

Type: Scalar Slot
Units: Fraction
Description: The maximum possible efficiency. It is also the default efficiency if there is no Depletion Requested
Must be between 0 and 1.0 inclusive.
I/O: Required Input
Supply Links: N/A

When Depletion Requested is known, Efficiency is computed as:

$$Efficiency[t] = min(\frac{DepletionRequested[t]}{Diversion[t]}, MaximumEfficiency)$$

If Depletion Requested is not known, Efficiency is equal to the Maximum Efficiency:

$$Efficiency[t] = MaximumEfficiency$$

Depletion is computed as:

$$Depletion[t] = Diversion[t] \times Efficiency[t]$$

A temporary variable tempReturnFlow[t] is computed:

$$tempReturnFlow[t] = (Diversion[t] - Depletion[t])$$

Then, the tempReturnFlow[t] is split and/or routed according to the selected method in the Return Flow Route or Split method.

Note: *See "Return Flow Route or Split" on page 36 for more information.*

Return Flow Route or Split

The Return Flow Route or Split category allows the user to specify if and how return flows will be split and/or routed. It has two methods: Simple Lag and Split and Route. When switching between these two methods, RiverWare must delete any existing supplies that link returnFlow or Returned Flows. If there is only one supply, a new supply will be created to the new slot. If there are multiple supplies, then no supplies will be re-created. Any input data on the supply WILL be lost. When you switch methods, a warning dialog will be posted to describe what will happen.

Simple Lag

This default method uses simple time lag that uses an integer number of timesteps.

+ returnFlow

Type:	Series Slot
Units:	Flow
Alt Units:	Volume
Description:	Amount of diverted water that is not consumed. This water then leaves the Diversion account as returnFlow
I/O:	Input, Rules, Object Level Accounting Method, or Account Solution

Supply Links: Yes, to Return Flow slot on a storage or passthrough account

+ Return Flow Lag

Type: Scalar Slot
Units: None (no of timesteps)
Description: The integer number of timesteps the returnFlow should be lagged from Diversion.
I/O: If not input, zero will be used
Supply Links: No

In general, the method will lag the internal tempReturnFlow[t] by the specified number of timesteps and set the returnFlow slot in the future. The full solution approach is presented in "Diversion Account Solution Equations" on page 44.

When any routing is considered, multiple dates (timesteps) are involved in the solution. Although tempReturnFlow[t] is a flow, it is converted to a volume in the computation to conserve mass. The result is converted back to flows for the lagged timestep. If the run timestep is a constant size, these flow/volume conversions are not needed nor performed as the length of t is the same as the length of the lagged timestep. The equations become much simpler.

$$returnFlow[t + lagTimesteps] = tempReturnFlow[t] \times \frac{TimestepLength[t]}{TimestepLength[t + lagTimesteps]}$$

Split and Route

Any return flow will have Multi Return Lag Coeffs. Then, the routed return flow will be set according to the coefficients.

Note: *The* Split and Route *method cannot be used by a diversion account in a computational subbasin that performs a water rights allocation using the prior appropriation method and water rights solver.*

+ Returned Flows

Type: Multi Slot (Accounting Multi Slot)
Units: Flow
Alt Units: Volume
Description: This slot stores the split and routed return flows.
As supply links are added, columns are added to this multi slot.
I/O: Output only
Supply Links: Yes, to Return Flow slot on storage or passthrough accounts.

+ Multi Return Lag Coeffs

Type: Table Slot
Units: Fraction

Description: There will be one column for each supply to Returned Flows.
The number of columns will be added automatically when the supply links are made. The number of valid values in each column determine the number of coefficients for that supply. These should sum to 1.0 down a column. Columns are not deleted if supplies are deleted.

I/O: Required Input.

Supply Links: No

+ PreRouted Return Flows

Type: Agg Series Slot

Units: Flow

Alt Units: Volume

Description: The return flows before they are routed
This slot is an Agg Series Slot but appears similar to a Multi Slot. If there are multiple supplies to Returned Flows, it has one column for the Total and then one column for each supply to Returned Flows. If there is only one supply, this slot only has one column with no total. The slot will be automatically resized when supplies are added or deleted.

I/O: Output Only during simulation, Input at pre-simulation timesteps if necessary

Supply Links: No

+ Return Flow Proportion

Type: Table Slot.

Units: Fraction

Description: This slot specifies how the split should be made. There is one column for each split/supply. The columns should be automatically added when the supply links to Returned Flows is made. Columns are not deleted if supplies are deleted.
The sum of the values should equal 1.0.

I/O: Required input

Supply Links: NA

The solution equation will set the PreRouted Return Flows slot for each column “i” (supply) based on the temporary value tempReturnFlow[t]:

$$PreRoutedReturnFlows[t, i] = tempReturnFlow[t] \times ReturnFlowProportion[i]$$

For each split, i.e. each column “i” of the Returned Flows slot and corresponding columns of the PreRouted Return Flows and Multi Return Lag Coeffs slot, the return flow is computed as:

$$returnedFlows[t, i] = (C_0 PreRoutedReturnFlows[t, i] \times ts(t) + C_1 PreRoutedReturnFlows[t-1, i] \times ts(t-1) + ... + C_{n-1} PreRoutedReturnFlows[t-n-1, i] \times ts(t-n-1) + C_n PreRoutedReturnFlows[t-n, i] \times ts(t-n)) / ts(t)$$

Where C is the appropriate coefficient in the Multi Return Lag Coeffs slot and n is the number of coefficients in the Multi Return Lag Coeffs slot. “ts()” indicates the timestep length of the specified timestep. In both cases, n is computed from the number of valid values in the appropriate column of the Multi Return Lag Coeffs slot.

Note: *Typically in the accounting system, routing sets the current value at an integer number of timesteps in the* ***future****. This method is different, here the equations are setting the current* Returned Flows *using values from previous timesteps. This method then requires that there are* PreRouted Return Flows *at t-n timesteps to solve. On pre-simulation timesteps, these should be input or set by a rule.*

Subordination of Right

This method allows an exception to the strict prior-appropriation priority determination to be respected by SolveWaterRights().

Subordinate Senior Rights

This method is used to specify that a downstream senior right may have to give up water in the event that this right would be shorted. More than one downstream senior right may be subordinated to this right, in which case, the subordinates give up water to satisfy this right in order of their relative priorities.

This method is used to specify that a downstream senior right may have to give up water in the event that this right would be shorted.

+ Subordinated Rights

Type:	List slot of Accounts
Units:	None
Alt Units:	NA
Description:	This list slot holds the set of downstream senior rights that are subordinated to this right.
I/O:	Input
Supply Links:	No

Water Right

Methods in this category specify whether the account has a water right.

None

No water right is defined for this account

Priority Right

When selected, this account will have a water right. Selecting this method is identical to checking the "Has Priority Date" toggle in the account configuration, General tab. The following slot is added when this method is selected:

+ Appropriation Request

Type:	Series slot
Units:	Flow
Alt Units:	Volume
Description:	A result of the water rights allocation solver. This is the allocatable flow that is the solver tries to allocate to this account.
I/O:	Set by Rule
Supply Links:	No

The value in the Appropriation Request slot does *not* reflect reductions in the allocation to prevent shorting downstream senior rights. It *does* reflect reductions based on legal constraints (such as minimum bypass) and physical constraints (such as the capacity of the physical diversion structure). For more information about the diversion capacity for the reach, "Diversion from Reach" in Objects.pdf.

+ Temp Reason

Type:	Series slot
Units:	None
Alt Units:	NA
Description:	This temporary slot encodes diagnostic information about the reason why the account is allocated the amount that it is given by the water rights solver.
I/O:	Output only
Supply Links:	NO

The water rights solver rule function, SolveWaterRights (or SolveWaterRightsWithLags) assigns a flow to a supply for a water right; this flow might be less than is requested in the Appropriation Request slot. The table below describes the various reasons the solver allocates what it does. Some of these values may be OR-ed together. This encoding is subject to change at any time.

Temp Reason	Allocation is limited by	Description
0000000001	Initial Request, which was input.	
0000000003	Initial Request, which was disaggregated by a subbasin.	
0000000005	Max Permitted method was selected.	No allocation is made. The Appropriation Request reflects the allocation that might have been made had this method not been selected.
0000000010	Min Bypass criteria	Min Bypass criteria are reflected in the Appropriation Request.
0000000040	Max Accrual	Not yet implemented.
0000000050	Max Legal Request	The initial request was larger than the Maximum Legal Request, so it was cutback to the Maximum Legal Request
0000000200	Diversion Capacity at source.	
0000000400	The available water in the supply chain.	This is the flow in the Allocatable Flow supply chain after higher-priority rights have been satisfied.
0000xxx000	A downstream senior placed a call.	A downstream senior's call further restricted the allocation (below the Appropriation Request). The priority number of the senior appears in these digits, except that if the priority is zero, the digits show 999.
0xxx000000	A junior placed a call on a senior subordinate.	The priority number of the calling junior appears in these digits. The junior cannot have priority zero.

+ Temp Available For Shared Priority

Type: Series slot
Units: Flow
Alt Units: Volume
Description: A result of the water rights allocation solver. When the Share Proportionally with Limits method is selected in the Account Equal Priority Allocation method category on the computational subbasin, this slot is instantiated. If this account has a priority date equal to other accounts this slot is populated with the amount of water available to this account independently of the other shared priority accounts at the time water is allocated. This value is used in the calculations that ultimately allocate water to the shared accounts as described under "Water Right Allocation" on page 167.
I/O: Rule
Supply Links: NO

Min Bypass

This method is executed as part of a prioritized water rights allocation. This method models a legal constraint that may reduce a water right's initial request for water by forcing the water right to leave a certain amount of water in the stream at a given reference location.

The Min Bypass methods are used to specify whether a minimum bypass is required when making water rights allocations.

None

No minimum bypass is defined for this account.

Fraction of Flow Above Min

The following slots are added for this method:

+ Bypass Reference Location

Type:	List
Units:	NA
Alt Units:	NA
Description:	Optional, control point at which you wish to keep a minimum bypass in the stream. If a control point is not specified in this slot, the accounting flow in the linked object will be used.
I/O:	Input only, only one control point can be specified
Supply Links:	No

+ Absolute Min Bypass

Type:	Periodic
Units:	TIME vs Flow
Alt Units:	NA
Description:	The required flow that must remain in the stream
I/O:	Input only
Supply Links:	No

+ Fraction above Min

Type:	Periodic
Units:	Time vs none
Alt Units:	NA
Description:	The fraction of the flow above the Absolute Min Bypass that must be left in the stream
I/O:	Input only
Supply Links:	No

The requirement contains two components, the Absolute Min Bypass and the Fraction above Min.

Following are three examples to show how this method calculates the flow:

- A legal requirement says to leave 300cfs in the stream. The following would be input: Absolute Min Bypass = 300cfs, Fraction above Min = 0.
- A legal requirement says to leave 45% of the flow in the stream. Input the following: Absolute Min Bypass = 0cfs, Fraction above Min = 0.45
- A legal requirement says to leave 300cfs plus 45% of the flow above that minimum in the stream. Input the following: Absolute Min Bypass = 300cfs, Fraction above Min = 0.45.

This method is used to specify the flow to leave in the supply chain. This location can be specified in two ways depending on whether a control point is specified in the Bypass Reference Location:

- If the there is a control point specified in the Bypass Reference Location, the accounting flows on that object will be used.
- If there is no control point specified in the Bypass Reference Location, then the accounting flows on the object supplying the allocation will be used. That is, for an allocation to a diversion account from reach, the amount of water available for appropriation from passthrough accounts on the reach will be used.

The required bypass flow is calculated using the:

- Sum of Temp Available for Appropriation slot for all passthrough accounts on the object (non-control points), or
- Sum of Outflow for all passthrough accounts at the control point reference location.

If the reference location is downstream, then any lags and/losses are applied so that the min bypass constraint includes enough water to meet the min bypass at the reference location. Then, the constraint (i.e. upper limit on the appropriation) is temporarily stored in the Temp Min Bypass Constraint slot on the diversion account but is not saved with the model file.

Max Legal Request

Methods in the category, Max Legal Request, allow you to specify the maximum legal request which then limits the Initial Request.

No Method

Default, no action method.

Max Request Series

The following slot is instantiated:

+ Maximum Request

Type: Series
Units: Flow
Description: This is the maximum legal flow that the diversion account can request. If not input, there is no maximum and the initial request is used directly.
I/O: Optional input

Supply Links: Not Linkable

The maximum request is applied at the same time as other legal constraints during the step when the solver computes the Appropriation Request. If there is a valid Maximum Request (not NaN), then:

$$\text{Appropriation Request}[date] = Min(\text{Initial Request}[date], \text{Maximum Request}[date])$$

Note: *If the Initial Request and the Maximum Request are less than zero, this method will set the Appropriation Request to zero.*

Note: *Additional legal and physical constraints are applied to compute the final Appropriation Request as described in* *"Computing Appropriation Request from Initial Request" on page 186.*

Diversion Account Solution Equations

Diversion accounts solve for Depletion and/or return flow when Diversion is known. Return flows may be lagged by an integral number of timesteps or using an impulse response approach, based on the selected Return Flow Route or Split method.

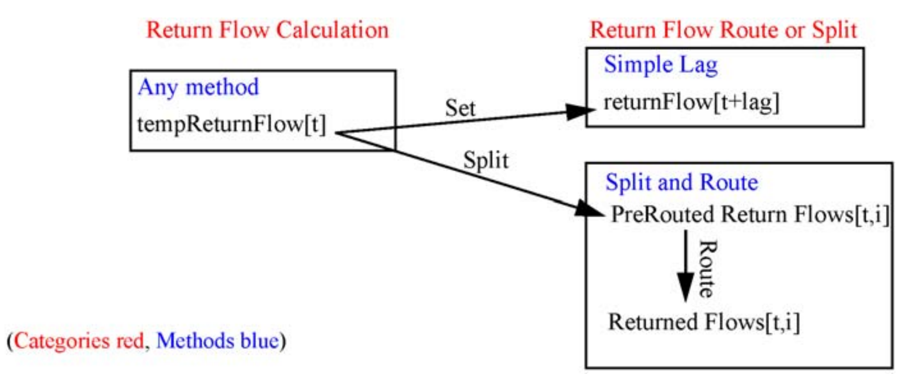
Required Knowns: Diversion

The Diversion Account has two categories whose methods influence the solution of the account:

- Return Flow Calculation
- Return Flow Route or Split

Figure1-6 shows the structure and the flow of information each selection:

Figure 1-6: Diversion Account structure



Thus, the solution of the diversion account proceeds as follows:

Execute the selected method in the Return Flow Calculation category. This will typically compute a temporary variable called tempReturnFlow[t]. These methods are described in “Return Flow Calculation” on page 32. If not already known, compute Depletion[t]:

$$Depletion[t] = Diversion[t] - tempReturnFlow[t]$$

Then, if it is not the Begin Accrual Date:

$$Accrual(t) = Accrual(t-1) + Depletion(t)(TimestepLength(t))$$

If it is the Begin Accrual Date

$$Accrual(t) = Depletion(t)(TimestepLength(t))$$

Execute the selected method in the Return Flow Route or Split category as described in “Return Flow Route or Split” on page 36. These methods take the tempReturnFlow[t] and route it and/or split it. The result is set in the returnFlow or Returned Flows slot.

Instream Flow Account

An Instream Flow account represents a legal instream flow right. Instream Flow accounts are only allowed on the control point object.

Instream Flow Slot Descriptions

Slot characteristics (Type, Units,...) are described in “Slot Descriptions” on page 13.

+ Accrual

Type:	Series Slot
Units:	Volume
Alt Units:	NA
Description:	Cumulative volume of inflows to the account from the Begin Accrual Date to current date.
I/O:	Account solution or Input (at the initial timestep)
Supply Links:	No

+ Flow

Type:	Series Slot
Units:	Flow
Alt Units:	Volume
Description:	Flow in the account used to meet the legal flow right. It is calculated as the sum of all inflows into accounts on this object

I/O: Account Solution
Supply Links: No

+ Inflow

Type: Multi Slot
Units: Flow
Alt Units: Volume
Description: Water transferred into the account from upstream
I/O: Propagated
Supply Links: Yes, from an Outflow on an upstream account

+ Maximum Accrual[1]

Type: Series Slot
Units: Volume
Alt Units: NA
Description: This slot represents the maximum accrual that is allowed during the accrual period. Maximum Accrual is not used in the account solution or in the water rights allocation but is available to hold data for viewing and use in Object Level Accounting Methods and Rules.
I/O: Input or Rules
Supply Links: No

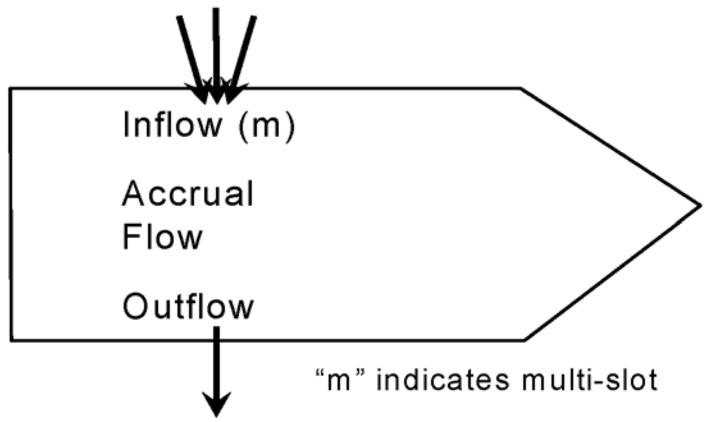
+ Outflow

Type: Series Slot
Units: Flow
Alt Units: Volume
Description: Water transferred out of this account downstream
I/O: Account Solution
Supply Links: Yes, to the Inflow of a downstream account

Figure1-7 is a diagram showing the general slots on the Instream Flow Account. The arrows leading into and out of the account represent supplies. The “m” indicate that the slot is a multi-slot and can be linked to more than one other slot and hence, there are multiple arrows.

1. Maximum Accrual is not yet used by SolveWaterRights().

Figure 1-7: General slots on the Instream Flow Account



Instream Flow Account User Method Categories

The instream flow account has the following user method categories:

- "Initial Request.”
- “Water Right” on page 49

Initial Request

This method is used to specify how the storage account calculates the Initial Request in the water rights allocation. The methods in this account are only valid if the account has a water right as specified by the Priority Right method.

None

No initial request is necessary, no appropriation is made, and no slots are added. This is the default method.

Specify Initial Request

When the Specify Initial Request method is selected, the following slots are added:

+ Initial Request

Type:	Series Slot
Units:	Flow
Alt Units:	Volume
Description:	The initial water rights appropriation request
I/O:	Required user Input or Rules
Supply Links:	No

+ Shortage

Type:	Series Slot
Units:	Flow
Alt Units:	Volume
Description:	The difference between the Initial Request and the Flow slot.
I/O:	Account Level Method
Supply Links:	No

This method allows the user to specify the initial appropriation request for use in a prioritized water rights allocation.

Based on Reference Level

When the Based on Reference Level method is selected, the following slots are added

+ Reference Level Stream Flow Table

Type:	Periodic
Units:	Time vs Flow
Alt Units:	NA
Description:	The periodic slot maps the reference level (as columns) from the underlying control point slot to an initial request (flow values in the table).
I/O:	Input Only
Supply Links:	No

+ Shortage

Type:	Series Slot
Units:	Flow
Alt Units:	Volume
Description:	The difference between the Initial Request and the Flow slot.
I/O:	Account Level Method
Supply Links:	No

This method is valid only if the Reservoir Storages Lookup method in the Instream Flow Reference Level category is selected on the containing control point. This method creates a slot called Reference Level which is an annual timestep slot whose value is computed once per year at the date indicated in the Start of Reference Year (also on the containing control point). For more information on the Reservoir Storages Lookup method, see "Reservoir Storages Lookup" in Objects.pdf.

This method is executed at the beginning of the timestep to lookup the current timestep and the Reference Level (on the control point) on the Reference Level Stream Flow Table to determine the Initial Request for the timestep.

Note: *If the current timestep is before the Start of Reference Year, the Reference Level used to compute the Initial Request is that of the prior year. This means that you must input an initial Reference Level* *if the Start of Reference Year does not coincide with the start of the run.*

Water Right

Methods in this category specify whether the account has a water right.

None

No water right is defined for this account.

Priority Right

When selected, this account will have a water right. Selecting this method is identical to checking the “Has Priority Date” toggle in the account configuration, General tab. The following slot is added when this method is selected:

Appropriation Request

Type: Series slot
Units: Flow
Alt Units: Volume
Description: A result of the water rights allocation solver. This slot gets the value of the initial request slot, since an instream flow account has no other legal or physical constraints that can reduce the appropriation request.
I/O: Rule
Supply Links: No

Available Allocatable Flow

Type: Series Slot
Units: Flow
Alt Units: Volume
Description: Upper bound on the amount of water an instream flow account can call from a junior account. An instream flow account is not allowed to call more than the allocatable flow in the channel, which quantity is saved in this slot by the water rights solver in SolveWaterRights when the solver’s controlling date is senior to this account, and is used by the water rights solver SolveWaterRights when the solver’s controlling date is junior to this account. The quantity saved in this slot is the outflow from the instream flow account’s sibling passthrough account having the water type given to the solver to identify the allocatable flow supply chain.
I/O: Rule
Supply Links: No

Temp Reason

Type: Series slot
Units: None

Alt Units: NA
Description: This temporary slot encodes diagnostic information about the reason why the account is allocated the amount that it is given by the water rights solver.
I/O: Output only
Supply Links: No

The water rights solver rule function, SolveWaterRights (or SolveWaterRightsWithLags) assigns a flow to a supply for a water right; this flow might be less than is requested in the Appropriation Request slot. The table below describes the various reasons the solver allocates what it does. Some of these values may be OR-ed together. This encoding is subject to change at any time.

Temp Reason	Allocation is limited by	Description
0000000001	Initial Request, which was input.	
0000000004	Initial Request, which was computed by Reference Level	
0000000040	Max Accrual	Not yet implemented.
0000000400	The available water in the supply chain.	This is the flow in the Allocatable Flow supply chain after higher-priority rights have been satisfied.
0xxx000000	A junior placed a call on a senior subordinate.	The priority number of the calling junior appears in these digits. The junior cannot have priority zero.
1000000000	A junior placed a call on a senior subordinate instream flow right.	

Instream Flow Account Solution Equations

The solution equations for the Instream Flow account is as follows:

The instream flow account always solves for the Flow slot regardless of knowns.

$$Flow = \sum \text{All Account Inflows on the Object}$$

To solve for Outflow and/or Accrual:

Required Knowns: Inflow

$$Outflow = Inflow$$

If it is not the Begin Accrual Date:

$$Accrual = Accrual(t-1) + Inflow \times TimestepLength$$

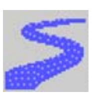
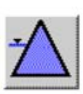
If it is the Begin Accrual Date

$$Accrual = Inflow \times TimestepLength$$

Passthrough Account

Passthrough accounts provide a means of transferring paper water between legal accounts. A user can look at any object in the model and see the paper water that is passing through that object at any given timestep.

Passthrough accounts can be created on reaches, bifurcations, confluences, diversion objects, control points, reservoirs, and stream gages. Depending on the type of object, passthrough accounts have different slots and therefore different components to the solution equation. The following table shows the slots that exist on a passthrough account on each of the possible objects.

	Object Type	Slots on Passthrough Accounts			
	Reach	• Inflow • Slot Inflow	• Outflow • Diversion	• Gain Loss • Return Flow	• Lag Time • Temp Available for Appropriation
	Bifurcation Confluence	• Inflow	• Outflow	• Slot Inflow	
	Control Point	• Inflow	• Outflow	• Slot Inflow	
	Diversion	• Inflow • Return Flow	• Outflow • Temp	• Slot Inflow • Temp Available for Appropriation	
	Storage Reservoir	• Inflow • Outflow • Slow Inflow	• Diversion • Return Flow	• Temp Available for Appropriation • Transfers In	• Transfer Out • Storage (Optional)
	Level Power Reservoir	• Inflow • Outflow • Slow Inflow	• Diversion • Return Flow	• Temp Available for Appropriation • Transfers In	• Transfer Out • Storage (Optional)
	StreamGage	• Inflow • Outflow • Slot Inflow	• Transfer In • Transfer Out		

Passthrough Account Slot Descriptions

Slot characteristics (Type, Units,...) are described in “Slot Descriptions” on page 13.

Diversion

Type:	Multi Slot
Units:	Flow
Alt Units:	Volume
Description:	Water leaving the passthrough account to a linked Diversion account
I/O:	Input, Rule, or Propagated
Supply Links:	Yes, to Diversion slot on a Diversion account

Gain Loss

Type:	Series Slot
Units:	Volume
Alt Units:	Flow
Description:	A loss or gain due to local effects including evaporation, seepage, precipitation. This gain or loss represents this account’s allocation of the physical loss on the simulation object. A gain is positive, a loss is negative.
I/O:	Input, Rules, Object Level Accounting Method, or Account Solution (if a gain loss coefficient is defined).
Supply Links:	No

Inflow

Type:	Multi Slot
Units:	Flow
Alt Units:	Volume
Description:	Water transferred into this account from upstream
I/O:	Propagated
Supply Links:	Yes, from an upstream account’s Outflow slot.

Lag Time

Type:	Scalar
Units:	time
Alt Units:	NA
Description:	Scalar slot defining the lag between net inflows (Inflow, Slot Inflow minus Diversion) and Outflow. See HERE (Section 2.4.3) for lag routing solution detail. Note, the water rights allocation solver requires that Lag Time be in integral number of timesteps.
I/O:	Optional Input
Supply Links:	No

+ Local Timestep Offset

Type:	Scalar
Units:	NONE
Alt Units:	NA
Description:	Scalar slot used by water rights allocation solver. This slot defines, in number of timesteps after the current run timestep, the timestep on which all water rights solver computations will be done for this account and the accounts it supplies.
I/O:	This slot value is computed, when necessary, by the computational subbasin that is used for a water rights allocation solution. It can be input, when desired, for accounts not set automatically. For more information on the use of this slot click HERE (Section 10.4.1).
Supply Links:	No

+ Outflow

Type:	Series Slot
Units:	Flow
Alt Units:	Volume
Description:	Water transferred out of this account downstream. Note, this is not a multi-slot. On a passthrough account, there can be zero or one outflow supply leading out of the account
I/O:	Typically Account Solution. On storage enabled reservoir account: Account Solution, Input, or Rule
Supply Links:	Yes, to a downstream account Inflow slot

+ Return Flow

Type:	Multi Slot
Units:	Flow
Alt Units:	Volume
Description:	Water entering the passthrough account as return flow from a diversion account
I/O:	Input, Rules, or Propagated
Supply Links:	Yes, to Return Flow slot on a Diversion account

+ Slot Inflow

Type:	Series Slot
Units:	Flow
Alt Units:	Volume
Description:	Water added to the account from a local source. This inflow represents this account’s allocation of the physical local inflows on the simulation object. If there is a non-zero Lag Time on the account, the Slot Inflow must be zero
I/O:	Input, Rules, or Object Level Accounting Method
Supply Links:	No

+ Storage

Type: Series Slot
Units: Volume
Alt Units: NA
Description: Storage balance in the account. The Storage slot is instantiated only if the "Allow Storage" toggle is selected in the account configuration. The solution algorithm when using storage is described further HERE (Section 2.4.3).
I/O: Input, Rules, or Account Solution
Supply Links: No

+ Temp Available For Appropriation

Type: Series Slot
Units: Flow
Alt Units: Volume
Description: Used by the water rights allocation solver to specify the amount available to allocate. Typically, this is the sum of the following account slots: inflows, slot inflows, transfers in, and storage, subtracting diversion and transfers out, each as applicable to the object on which the passthrough resides. For example, storage is only applicable on storage or level power reservoirs for which "Allow Storage" is activated. It is the Volume in the account before loss and lag are applied and before return flows are added. Temp Available For Appropriation is a temporary slot and is not stored in the model file.
I/O: Account Solution
Supply Links: No

+ Transfers In

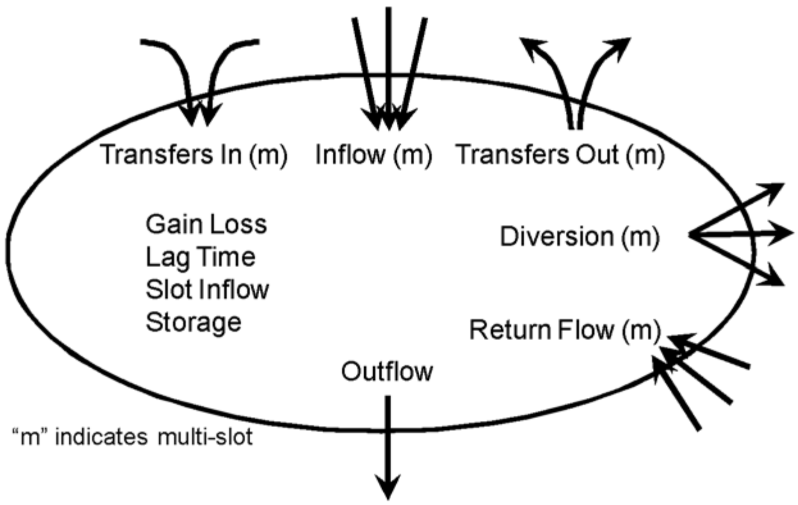
Type: Multi Slot
Units: Flow
Alt Units: Volume
Description: Transfers into the account from an account on the same object. This slot is only available on gage and reservoir objects.
I/O: Input, Propagated, or Rules
Supply Links: Yes, from a Transfers Out slot on another account on the same object

+ Transfers Out

Type: Multi Slot
Units: Flow
Alt Units: Volume
Description: Transfers out of this account to an account on the same object. This slot is only available on gage and reservoir objects.
I/O: Input, Propagated, or Rules
Supply Links: Yes, to a Transfers In slot on another account on the same object

Figure1-8 is a diagram showing the slots on the passthrough account. The arrows leading into and out of the account represent supplies. The "m" indicate that the slot is a multi-slot and can be linked to more than one other slot and hence, there are multiple arrows.

Figure 1-8: Slots on the passthrough account



Passthrough Account User Methods

There user selectable *Account* Level Methods on passthrough accounts:

- "Gain Loss Coefficient."
- "Outflow Limitation" on page 56.
- "Negative Flow Appropriation" on page 57.

For information on the compiled and user defined *Object* Level Accounting Methods available on passthrough accounts, see "Object Level Accounting Methods" on page 66.

Gain Loss Coefficient

This category is available on reaches. Methods in this category determine the type of slot in which gain loss coefficients are entered.

Constant Gain Loss Coefficient

Under this method, a scalar slot is added where a single non-time varying gain loss coefficient can be specified.

+ Gain Loss Coefficient

Type: Scalar
Units: Decimal
Alt Units: NA
Description: If defined, this scalar value is multiplied by the passthrough account's flow (routed Inflows plus Slot Inflow minus Diversion) to calculate Gain Loss

I/O: Optional Input
Supply Links: No

Variable Gain Loss Coefficient

Selecting this method provides a series slot for the entry of gain loss coefficients.

+ Variable Gain Loss Coefficient

Type: Series
Units: Decimal
Alt Units: NA
Description: Coefficients applicable to individual timesteps can be entered. If defined for the timestep, the coefficient value is multiplied by the passthrough account's flow (routed Inflows plus Slot Inflow minus Diversion) to calculate Gain Loss
I/O: Optional Input
Supply Links: No

Periodic Gain Loss Coefficient

Under this method, a periodic slot is provided for the entry of gain loss coefficients.

+ Periodic Gain Loss Coefficient

Type: Periodic
Units: Decimal
Alt Units: NA
Description: Coefficients can be entered by the periods specified in the slot. If defined for the period, the coefficient value is multiplied by the passthrough account's flow (routed Inflows plus Slot Inflow minus Diversion) to calculate Gain Loss. By default, the slot is configured to "Lookup" can be configured to "Interpolate"
I/O: Optional Input
Supply Links: No

Outflow Limitation

This category is available on reaches.

None

No outflow limitation is defined.

Scalar Max Outflow

The passthrough will limit the outflow to the max outflow and send the remaining water to the diversion supply. One sample use of this method is when you want to send all of the water in a supply chain to a single diversion account. You can set the Max Outflow to zero. There is no need to set the diversion supply as it will be set when the passthrough solves.

+ Max Outflow

Type: Scalar
Units: Flow
Description: The maximum that can flow through the Outflow supply.
I/O: Required Input or error
Supply Links: Not Linkable

When this method is selected, the passthrough account solution equations limit the Outflow to the Max Outflow. Remaining water goes to the linked diversion supply. The solution equations are described in “Equation if the Scalar Max Outflow method is selected” on page 61.

There are a few limitations with this method:

- There must be exactly one Diversion supply coming off the passthrough account. This will be checked at run time.
- The Max Outflow method only applies to a single passthrough account, not the entire object.
- This method can be used in a SolveWaterRights algorithm but the linked diversion supply cannot be used to meet a water right. In that case, the passthrough account would be overdetermined as both the solver and this account would be trying to set the diversion supply.

Negative Flow Appropriation

This category is available on reaches, reservoirs and control points. It affects how the water rights solver deals with negative flows. See Chapter 5, “Water Right Allocation,” on page 167.

None

At the time of allocation, any negative flows will lead to an error.

Allow Negative Flows

The passthrough account method (Allow Negative Flows) will allow the user to specify that negative flows are allowed in the supply chain during water right allocation. It will also add the following slot necessary for the proposed changes.

+ Temp Available Before Appropriation

Type: Series
Units: Flow
Description: The flow available for appropriation before visiting a water right.
This slot is only used within the water rights solver; it is not returned to the calling rule. As a result, it will never show any values in it and could be made invisible to the user (if that were supported on the Open Account dialog).
I/O: Output only
Supply Links: Not linkable

Passthrough Account Solution Equations

The solution equations used for passthrough accounts are determined by the type of object and the configuration of the account. First we discuss the general equation, then look at passthrough accounts with lags and losses, look at passthrough accounts where storage is allowed and finally look at passthrough accounts with an outflow limitation.

General Equation

The components of the equation depend on the type of object on which the passthrough account resides. For example, a passthrough account on a confluence, bifurcation or control point only has the Outflow, Inflow, and Slot Inflow components. See the table in “Passthrough Account” on page 51 for the list of the slots on passthrough accounts for each object.

Required Knowns: Slot Inflow; other unknown variables in the equation will default to zero.

The equation to calculate Outflow is:

$$Outflow = Inflow + SlotInflow + \frac{GainLoss}{TimestepLength} - Diversion + ReturnFlow + TransfersIn - TransfersOut$$

The solution becomes more complicated if the account is on a reach and there are lags or calculated losses or if the account is on a reservoir and storage is allowed. These solutions are described below.

Equations for Passthrough accounts with Gain Loss Coefficients and/or Lag Times

Passthrough accounts on Reach objects solve slightly differently if the a gain loss coefficient is defined or the Lag Time is non-zero:

- First, the gain loss coefficient, tempGainLossCoefficient(t) is obtained from the selected Gain Loss Coefficient method. See “Gain Loss Coefficient” on page 55.
- If the tempGainLossCoefficient is non-zero (and Lag Time is NaN), Gain Loss is calculated as a function of the inflows. Remember that Gain Loss is a volume. The account is then able to solve for Outflow using the general equation above.

$$GainLoss = tempGainLossCoefficient(t) \times (Inflow + SlotInflow - Diversion) \times TimestepLength$$

- If there is a non-zero Lag Time, the solution equations account for the lag. Note, if there is a lag defined, Slot Inflow must be zero or an error occurs.

First, if the lag is an integer number of timesteps:

Required Knowns: Slot Inflow(t) to solve for Outflow(t+lag);

If the tempGainLossCoefficient is non-zero:

$$GainLoss(t+lag) = tempGainLossCoefficient(t) \times (Inflow(t) + SlotInflow(t) - Diversion(t)) \times TimestepLength(t)$$

Otherwise, GainLoss(t+lag) may be specified (via Input, Object Level Accounting Method, or Rules) or it defaults to zero. The Inflow, Slot Inflow (must be zero) and Diversion all occur before the lag, Gain Loss and Return flow occur after the lag(i.e. on the lagged timestep). Outflow is calculated as:

$$Outflow(t+lag) = Inflow(t) + SlotInflow(t) + \frac{GainLoss(t+lag)}{TimestepLength(t)} - Diversion(t) + ReturnFlow(t+lag)$$

If the lag is not an integer number of timesteps:

Required Knowns: Slot Inflow(t) and Inflow(t-1) to solve for Outflow(t+lagInt);

The lag is broken up into an integer number of timesteps (lagInt) and two fractions (lagFrac1 and lagFrac2) representing the flow that occurs on the two adjacent timesteps.

$$lagInt = IntegerValueOf\left(\frac{LagTime}{timestep}\right)$$

$$lagFrac1 = 1 - \left(\frac{LagTime}{timestep} - lagInt\right)$$

$$lagFrac2 = 1 - lagFrac1$$

If the tempGainLossCoefficient(t) is non-zero:

$$GainLoss(t+lagInt) = tempGainLossCoefficient(t) \times TimestepLength(t) \times (lagFrac1 \times Inflow(t) + lagFrac2 \times Inflow(t-1) + SlotInflow(t) - Diversion(t))$$

Otherwise, GainLoss(t+lag) must be specified (via input, Object Level Accounting Method, or rules) or it defaults to zero. The Outflow is then calculated as:

$$Outflow(t + lagInt) = lagFrac1 \times Inflow(t) + lagFrac2 \times Inflow(t-1) + SlotInflow(t) - Diversion(t) + \frac{GainLoss(t + lagInt)}{TimestepLength(t)} + ReturnFlow(t + lag)$$

Equation if Storage is allowed

If the passthrough account is on a reservoir object, then the user has the option to allow storage. Storage is only allowed when the user enables the “Allow Storage” toggle in the account configuration. Storage is intended to be used on a temporary basis. If outflow is assigned by the user or a rule, a storage value may be calculated and set in the storage slot for a given timestep. On the next timestep, the account solves to try to make storage equal to zero. The purpose of allowing storage is that some reservoirs have constraints that limit the total water that can be released. Because this constraint limits outflows, accounting water must be stored until it can be physically released from the reservoir. The following procedure is used. Note, in this case, Inflow must be known; it does not default to zero. Also note, unlike a reach, a lag or gain loss coefficient is not allowed on a reservoir passthrough account. Therefore all calculations occur at a given timestep (except where indicated) and Gain Loss cannot be calculated by an account level method (although it can be calculated by an object level accounting method).

Required Knowns: Inflow and Slot Inflow; other unknown variables in the equation will default to zero.

- If both Storage and Outflow are unknown, Storage remains unknown and Outflow is calculated. This is similar to the general equation except the entire previous storage is released.

$$Outflow = \frac{Storage(t-1)}{TimestepLength} + Inflow + SlotInflow + \frac{GainLoss}{TimestepLength} - Diversion + ReturnFlow + TransfersIn - TransfersOut$$

- If Outflow is known and Storage is unknown, Storage is calculated:

$$Storage = Storage(t-1) + GainLoss + TimestepLength \times (Inflow - Outflow + SlotInflow - Diversion + ReturnFlow + TransfersIn - TransfersOut)$$

- If Storage is known and Outflow is unknown, Outflow is calculated:

$$Outflow = Inflow + SlotInflow + \frac{GainLoss}{TimestepLength} - Diversion + ReturnFlow + TransfersIn - TransfersOut + \left(\frac{Storage(t-1) - Storage}{TimestepLength}\right)$$

- If both Storage and Outflow are known, an error is issued.

Equation if the `Scalar Max Outflow` method is selected

On a reach passthrough account with the `Scalar Max Outflow` method selected (in the `Outflow Limitation` category), the following equation is used:

$$Outflow = Min(Inflow + SlotInflow, MaxOutflow)$$

Required Knowns: Slot Inflow; other unknown variables in the equation will default to zero.

The remaining water is set on the diversion supply.

$$diversion\ supply = Inflow + SlotInflow - Outflow = Diversion$$

There must be exactly one diversion supply. Therefore, the Diversion slot value is identical to the value in the diversion supply. Lag, gain loss, and return flow are NOT allowed. This is checked at run time.

Properties of Accounts

In addition to the slots that reside on each account, there are variables that can be used to describe the account including Water Type and Water Owner. These properties are used to keep accounts clearly identified and, as we will see later, used by predefined rule functions to access the correct account.

Water Type

Water Type is a property of an account that can be used to describe that account. Typically, Water Type is used to classify where the water originated or where it is going. For example, in a model with a trans-basin diversion, the user could define two water types, Type A and Type B, one for each basin. When water is diverted trans-basin, the Water Type can be used to track water from basin A when it is in basin B. The default Water Type is NONE.

Water Owner

Water Owner is another property of accounts used to classify or describe an account. Typically, the Water Owner is used when a volume of water is owned by a specific user who wishes to track that water through a basin. For example, a city might own a set volume of water in an upstream reservoir. The default Water Owner is NONE.

By defining both Water Type and Water Owner, the user can track the classification of a wide range of water. If we combine the two examples above, the city might own some of the trans-basin water but cannot use it until it is diverted from Basin A to Basin B and flows downstream to the city. By defining both Water Type and Water Owner, the city can track that the water released for the city (which is located in Basin B) is from Basin A and owned by the city.

The Water Owner and Water Type are defined in the Accounting System Configuration and apply to a specific model. When configuring an individual account, the default Water Owner and Water Type is NONE but can be changed by the user.

Priority Date

Legal accounts (Storage, Diversion, Instream Flow) can have a priority date associated with the account. Specified in the account's configuration dialog, the priority date is a unique fully specified date and time. If the Water Rights method is selected on the Account, then a Priority Date must be specified and vice versa. The Priority Date is used by the water rights allocation solver to allocate flow to the account. For more information, see Chapter 5, "Water Right Allocation," on page 167.

Supplies

Accounts, defined on the simulation objects, track water as it moves through the network. Accounts are connected to other accounts by an object called a Supply.

> Supply
>
> A supply links two slots on two accounts. Specifically, a supply to an account means that paper water is transferred into that account from another (often upstream) account.

The term supply comes from the relationship between the two accounts. The account on the upstream end of the link *supplies* the downstream account with water. Similarly, the downstream account has a *demand* on the upstream account. When looking at a given account, there may be supplies coming into the account and demands leaving the account. We can also say that there are supplies to the accounts and that the account has demands. The demand to one account is also considered a supply to the downstream account. As a result, we refer to all of the links in the accounting system as Supplies.

Supplies are workspace objects, like a Storage Reservoir or Reach simulation objects. Because they are workspace objects, supplies are referenced by RPL using a different syntax than simulation slots. This syntax used to reference a supply is “SupplyName.Supply”. This is described in more detail in later chapters.

Typically, there are multiple supplies both entering and leaving an account. Supplies can connect two slots on the accounts on the same simulation object or on other simulation objects as long as there is a physical simulation link between the two objects. In addition, there may be multiple supplies connecting the same two accounts.

Supply Names

By default, the supply name indicates the “from” object, the “from” account, the text “t”, the “to” object, and “to” account. For Diversion/Return Flow and Transfer supplies (see next section), the string “Div” and “Tran” are appended, respectively.

The format for supply names can be controlled by the user in the Supply Name Format dialog. (See “Supply Name Format” on page 129.) The format specified in this dialog is applied to new supplies and can be applied to existing supplies from the Supply Manager. The user can also individually set supply names from the open account dialog Supplies tab. See “Configuring Accounts through the Open Account dialog” on page 101 for more information.

Types of Supplies

When a supply is created, the user must select how the supply is going to be used. In the supply creation menu, the user can select one of the following supplies classifications. The selected type determines which slots will be linked. Note, a link in the physical or simulation system must exist to create a supply between two accounts.

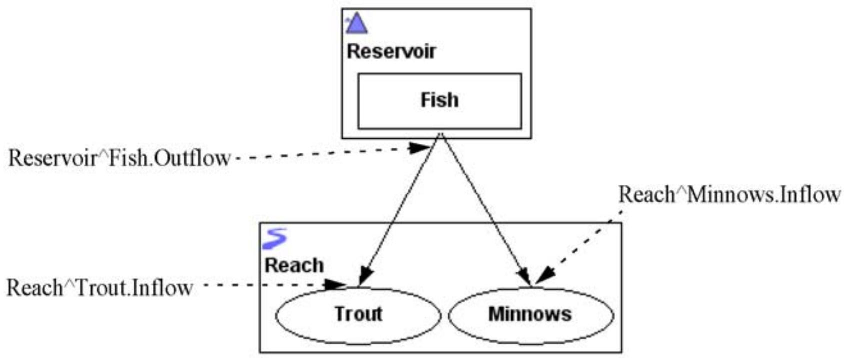
Inflow/Outflow

A supply connecting an inflow on an account on one object and an Outflow on an account on another object. These are used to represent the water that is moving from one object to another as an outflow to an inflow. The supply is created from the downstream account to the upstream account.

Example: On the diagram, an Inflow/Outflow supply connects:

Reservoir^Fish.Outflow to Reach^Trout.Inflow and Reservoir^Fish.Outflow to Reach^Minnows.Inflow

Figure 1-9: Inflow/Outflow example



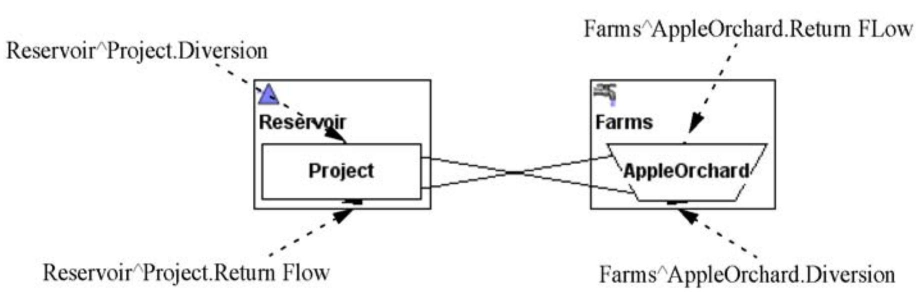
Diversion/Return Flow

A supply connecting a Diversion (or Return Flow) on an account on one object to a Diversion (or Return Flow) on an account on another object. These supplies are used to represent water that is diverted from one object to another. They are also used to represent return flows from one object to another. A supply connecting Diversion slots is created starting at the receiving object, typically a water user. A supply connecting Return Flow slots is also created by starting at the receiving object, in this case, it is usually the reach or reservoir object.

Example: On the diagram, a Diversion/Return Flow supply connects:

Reservoir^Project.Diversion to Farms^AppleOrchard.Diversion and Farms^AppleOrchard.ReturnFlow to Reservoir^Project.ReturnFlow

Figure 1-10: Diversion/Return Flow example



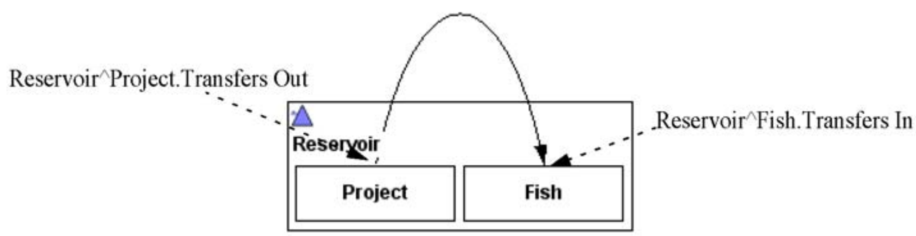
Transfer

A supply connecting a Transfer In on an account to a Transfer Out on another account on the same object. These supplies are used to represent water that is transferred from one account to another on the same object. The supply is always created starting at the receiving account, the one that is receiving the Transfer In.

Example: On the diagram, a Transfer type supply connects:

Reservoir^Project.Transfers Out to Reservoisr^Fish.Transfers In

Figure 1-11: Transfer type supply example



Properties of Supplies

Accounts have optional Water Type and Water Owner properties that can be used to classify and label the accounts. Similarly, Supplies have properties that can be used to label the purpose and/or classify the account. These optional properties are called Release Type and Destination Type and are described as follows:

Release Type

Supplies can be labeled according to their release type. For example, a release may be made for environmental flow purposes.

Destination Type

In addition, supplies can be labeled according to their Destination Type. For example, a release may be made from a reservoir to be used at a downstream water user.

Thus, using the Release Type and Destination Type properties, the user can specify both the reason for releases and where the water is ultimately headed. For example, a release might be made for flood control purposes but can also be diverted downstream for use in a groundwater recharge structure.

The Release Type and Destination Type are defined in the Accounting System Configuration and apply to a specific model. When configuring an individual supply, the default Release Type and Destination Type is NONE but can be changed by the user.

Setting Values with a DMI

If a DMI is used to bring in values to account inflows, outflows or transfers, it should set supplies in the accounting system rather than accounting slots. This is similar to the concept of setting supplies with rules as described in "Setting Slots versus Setting Supplies" on page 83.

Object Level Accounting Methods

In simulation, local inflows and gains/losses are typically input by the user, set by rules, or calculated by methods on the objects. For example, local inflows to a reservoir may be input on the Hydrologic Inflow slot and daily Evaporation may be calculated from a monthly table. These components represent a change in the physical amount of water in the object and must also be represented by the accounting system.

A structure, called Object Level Accounting Methods, are used to allocate physical gains and/or losses to the accounting system. They can also be used to reconcile the physical and accounting systems. They are called "object level" because they allocate water from a given simulation object to one or more accounts on that object.

> Object Level Accounting Methods
>
> Object Level Accounting methods are used to allocate physical gains, losses, and local inflows on an object to accounting slots on that object. In addition, they can be used to reconcile the physical and accounting system.

There are two types of Object Level Accounting Methods, *Compiled* and *User Defined*. See "Compiled Accounting Methods" and "User Defined Accounting Methods" on page 70.

Compiled Accounting Methods

> Compiled accounting methods
>
> Compiled accounting methods are hard-coded Object Level Accounting Methods. These methods were created by CADSWES and are available in a library. They represent either commonly used functionality or methods that are too complex to represent as user defined methods.

Examples of compiled accounting methods include commonly used methods like the Zero Slot Inflow method in the Reservoir Account Slot Inflow category which assigns zero to each account's Slot Inflow. In addition, there are complex methods that were too difficult to implement in RPL. For example, there is the Heron Inflow Calculation; a method specific to basins that have Rio Grande water types. The following table briefly describes each of the compiled methods (the default method for each category is shown in **orange**; the general methods are also described in more detail after the table):

(Obj)Category	Method	Description
Agg Diversion Account Reconciliation	**No Method**	No action - never executed
Bifurcation Account Slot Inflow	No Method	No action - never executed
	Zero Slot Inflows	Sets all Slot Inflows to 0.0

(Obj)Category	Method	Description
Confluence Account Slot Inflow	No Method	No action - never executed
	Zero Slot Inflows	Sets all Slot Inflows to 0.0
	Rio Grande Inflow 2[a]	Confluence^RioGrande.Slot Inflow is set equal to the Inflow 2. All other accounts' Slot Inflow set to 0.0
	Sidewater Inflow 2[a]	Distribute Confluence.Inflow 2 to the Floriston Rate and Undes account Slot Inflow.
Control Point Account Slot Inflow	No Method	No action - never executed
	Zero Slot Inflows	Sets all Slot Inflows to 0.0
	Copy Slot to Slot Inflows	On the specified account, set the Slot Inflow equal to the Local Inflow. Set all other Slot Inflows to 0.0. See "Copy Slot to Slot Inflows" on page 69
Distribution Canal Account Slot Inflow	No Method	No action - never executed
	Zero Slot Inflows	Sets all Slot Inflows to 0.0
Diversion Object Account Slot Inflow	No Method	No action - never executed
	Zero Slot Inflows	Sets all Slot Inflows to 0.0
Inline Pump Account Slot Inflow	No Method	No action - never executed
	Zero Slot Inflows	Sets all Slot Inflows to 0.0
Pipe Junction Account Slot Inflow	No Method	No action - never executed
	Zero Slot Inflows	Sets all Slot Inflows to 0.0
Pipeline Account Slot Inflow	No Method	No action - never executed
	Zero Slot Inflows	Sets all Slot Inflows to 0.0
Reach Account Slot Inflow	No Method	No action - never executed
	Zero Slot Inflows	Sets all Slot Inflows to 0.0
	Reconcile Rio Grande Outflow[a]	Reach^RioGrande.Slot Inflow is equal to the Reach.Outflow minus sum of WT(SanJuan).Outflow minus RioGrande.Inflow. Reach^WT(SanJuan).Slot Inflow = 0[b]
	Rio Grande Local Inflow[a]	Reach^RioGrande.Slot Inflow is set equal to the Reach.Local Inflow. All other accounts' Slot Inflows are set to 0.0
	Provo River Local Inflow[a]	Reach^ProvoRiver.Slot Inflow is set equal to the Local Inflow.
	NIC Local Inflow[a]	Reach^NIC.Slot Inflow is set equal to the Local Inflow.
	Copy Slot to Slot Inflows	On the specified account, set the Slot Inflow equal to the Local Inflow. Set all other Slot Inflows to 0.0. See "Copy Slot to Slot Inflows" on page 69 for more information.
Reach Account Gain Loss	No Method	No action - never executed
	San Juan Gain Loss[a]	Sets Gain Loss on accounts with water types Rio Grande and San Juan. Many accounting slots are registered as dependencies.

(Obj)Category	Method	Description
Reservoir Account Slot Inflow Reservoirs: • Storage • Level Power • Sloped Power • Pumped Storage	**No Method**	No action - never executed
	Zero Slot Inflows	Sets all Slot Inflows to 0.0
	Heron Inflow Calculation	Sets Slot Inflow on accounts with water types Rio Grande and San Juan. Many accounting slots are registered as dependencies.
	Pooled Account Slot Inflow[a]	Pooled.SlotInflow is set equal to the object's Hydrologic Inflow Net
	Donner Inflow[a]	Basin specific
	Prosser Uncomm[a]	Basin specific
	Copy Slot to Slot Inflows	On the specified account, set the Slot Inflow equal to the Hydrologic Inflow. Set all other Slot Inflows to 0.0. Click "Copy Slot to Slot Inflows" on page 69
Reservoir Account Gain Loss Reservoirs: • Storage • Level Power • Sloped Power • Pumped Storage	**No Method**	No action - never executed
	Heron Gain Loss Calculation[a]	Sets Gain Loss on accounts with water types Rio Grande and San Juan. Many accounting slots are registered as dependencies.
	El Vado Loss Calculation[a]	
	Nambe Falls Loss Calculation[a]	
	Elephant Butte Loss Calculation[a]	
	Elephant Butte Loss with RG Compact[a]	Sets Gain Loss on accounts with water types Rio Grande and San Juan Includes logic for the RG Compact. Many accounting slots are registered as dependencies.
	Abiquiu Loss Calculation[a]	Sets Gain Loss on accounts with water types Rio Grande and San Juan. Many accounting slots are registered as dependencies.
	Jemez Loss Calculation[a]	
	Cochiti Loss Calculation[a]	
Reservoir Account Reconciliation	**No Method**	No action - never executed
Water User Account Reconciliation	**No Method**	No action
Stream Gage Account Slot Inflow	**No Method**	No action - never executed
	Zero Slot Inflows	Sets all Slot Inflows to 0.0
	Reconcile Rio Grande Outflow[a]	StreamGage^RioGrande.Slot Inflow is equal to the StreamGage.Outflow minus Sum of WT(SanJuan).Outflow minus RioGrande.Inflow. StreamGage^WT(SanJuan).Slot Inflow = 0<times 10pt> [b] Many accounting slots are registered as dependencies.
	All Rio Grande	Sets Slot Inflow on accounts with water types Rio Grande
	All San Juan Chama[a]	Sets Slot Inflow on accounts with water types San Juan

a. Contact riverware-support@colorado.edu for more information on these basin specific methods.
b. WT(<name>) indicates all accounts with Water Type <name>)

The following are the description of the generally and commonly used compiled methods. Each method is executed according to its selected execution time as described “Reconciliation” on page 75. The default execution time for each method is also listed.

No Method

The default No Method is a no-action method and does nothing. In fact, the method has a static execution time of “Never” meaning it will never execute.

Zero Slot Inflows

The Zero Slot Inflows method on many of the object’s account slot inflow category sets the Slot Inflow to 0.0 for each account. By default, it is executed once at the beginning of the run, but the user can change this if desired.

Copy Slot to Slot Inflows

The Copy Slot to Slot Inflows method copies the object’s local inflow to the target account’s Slot Inflow and sets the other accounts’ Slot Inflow to zero.

Selecting the Copy Slot to Slot Inflows method instantiates the following slot:

+ Target Account

Type: List Slot
Units: No Units
Description: This slot contains one account on the object. This account’s Slot Inflow will be set equal to the local inflow.
Only one account is allowed in this list and the account must be on the same object as this slot. Also, the account must be a storage or passthrough account.
This slot can be set (by account name, water type, or water owner) for many objects at once using the Multiple Object Method Selector HERE (ObjectDialogs.pdf, Section 4).
I/O: Required Input
Supply Links: No

The method is available only in the <Object> Account Slot Inflow category on the following objects with one of the specified local inflow methods selected.

Object	Local Inflow Category	Possible methods	local inflow slot
Control Point	Local Inflow	Input Local Inflow	Local Inflow
Reach	Local Inflow and Solution Direction	Specify Local Inflow, Solve Inflow or Outflow Specify Local Inflow, Solve Outflow Solve Inflow, Outflow or Local Inflow Contingent Local Inflow or Solve Outflow	Local Inflow

Object	Local Inflow Category	Possible methods	local inflow slot
Reservoirs: Storage, Level Power	hydrologicInflow CalculationCategory	solveHydrologicInflow inputHydrologicInflow Hydrologic Inflow and Loss	Hydrologic Inflow
Reservoirs: Sloped Power, Pumped Storage	hydrologicInflow CalculationCategory	inputHydrologicInflow	Hydrologic Inflow

The Copy Slot to Slot Inflows method will first assign zero to the current timestep value of the Slot Inflow slot of ALL accounts on the object. It will then copy the current timestep value in the object's "local inflow slot" (as shown in the above table) to the specified Target Account's Slot Inflow slot. By default this method will be given an execution time of Beg of Timestep Once meaning it will be executed once at each timestep before simulation starts. This execution time can be changed by the user.

User Defined Accounting Methods

Alternatively, User Defined Accounting Methods are created by the user in the RiverWare Policy Language (RPL).

User Defined Accounting Methods

User Defined Accounting Methods are Object Level Accounting Methods created by the user in the RiverWare Policy Language.

For example, User Defined Accounting Methods can be used to specify how Hydrologic Inflow on a reservoir is allocated to the accounts on that reservoir or how the seepage in a reach is charged to the passthrough accounts on the reach. Often the methods depend on policy decision and are not physically based. For example, seepage in a reach may be charged only to the allocatable flow for that reach, not to any of the other water released from upstream reservoirs for downstream diversions. As a result, the methods are basin specific and must be created for each basin.

Because the methods are basin specific, the user must define and configure these methods for each basin. On storage and passthrough accounts, there are slots called Slot Inflow and Gain Loss. These are the slots in the accounting system that are set by the User Defined Accounting Methods to allocate local inflows and apportion gains and losses, respectively. This is done in the RiverWare Policy Language (RPL) using the Accounting Method Set Editor. In this editor, there are pre-configured policy groups for the allowed Objects/Accounts and the intended actions. For example, there is a policy group for the Reservoir Account Slot Inflow and Reservoir Account Gain Loss. These policy groups correspond to categories on the appropriate object.

Example

In DeepLake, the Hydrologic Inflow is shared equally amongst three accounts in a Deep Lake: A, B, and C. The Storage Account Slot Inflow method would be the following:

- DeepLake ^ "A.Slot Inflow"[] = DeepLake."Hydrologic Inflow"[] / 3
- DeepLake ^ "B.Slot Inflow"[] = DeepLake."Hydrologic Inflow"[] / 3
- DeepLake ^ "C.Slot Inflow"[] = DeepLake."Hydrologic Inflow"[] / 3

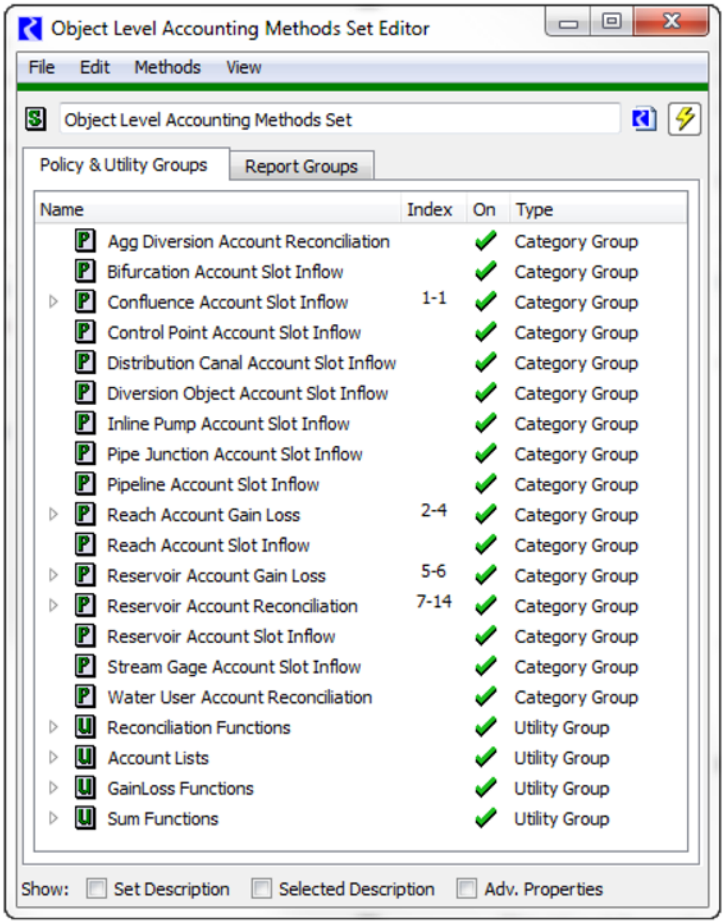
When the user creates a new Method in one of these policy groups, the new method appears as a method in the appropriate category on the object's Accounting Methods tab and can be selected by the user. By selecting this method on the object, the user is telling the object that the selected User Defined Accounting Method should apply to the accounts on that object. As a result, the method created for each object applies to all of the accounts on the object. Multiple assignment statements may be necessary to set all necessary slots.

The following table lists the policy groups / categories in the Accounting Method Set Editor and the associated objects and the accounts to which they apply

Policy Group / Category	Object	Account(s)
Agg Diversion Account Reconciliation	Aggregate Diversion	Diversion
Bifurcation Account Slot Inflow	Bifurcation	Passthrough
Confluence Account Slot Inflow	Confluence	Passthrough
Control Point Account Slot Inflow	Control Point	Passthrough
Distribution Canal Account Slot Inflow	Distribution Canal	Passthrough
Diversion Object Account Slot Inflow	Diversion Object	Passthrough
Inline Pump Account Slot Inflow	Inline Pump	Passthrough
Pipe Junction Account Slot Inflow	Pipe Junction	Passthrough
Pipeline Account Slot Inflow	Pipeline	Passthrough
Reach Account Gain Loss	Reach	Passthrough
Reach Account Slot Inflow	Reach	Passthrough
Reservoir Account Gain Loss	Storage Reservoir Level Power Reservoir Pumped Storage Reservoir Sloped Power Reservoir	Storage and/or Passthrough
Reservoir Account Reconciliation	Storage Reservoir Level Power Reservoir Pumped Storage Reservoir Sloped Power Reservoir	Storage and/or Passthrough
Reservoir Account Slot Inflow	Storage Reservoir Level Power Reservoir Pumped Storage Reservoir Sloped Power Reservoir	Storage and/or Passthrough
Stream Gage Account Slot Inflow	Gage	Passthrough
Water User Account Reconciliation	Water User	Diversion

User Defined Accounting Methods, although written in RPL, do not behave the same as rules. Although the methods are prioritized in the method set editor, the priority is not used. Instead, there can only be one method selected for each object. The methods execute according to their specified execution time. For more information, see "Reconciliation" on page 75.

Figure 1-12: Screenshot of the Object Level Accounting Method Set Editor



Although the methods are specific to a basin, there are a few features that can be used to generalize the methods. The keyword *ThisObject* (note, there are no space) can be used in place of a specific object name. When called, it will replace this with the name of the object from which the method is called.

Example:

Following is a reach Pass Through Gain Loss method that makes use of the "ThisObject" syntax. It would be useful on this basin because each reach has two accounts, Fish and Farmers. The Fish account gets charged with all loss, Farmers get none. Each reach in the model could then use this method.

ThisObject ^ "Fish.Gain Loss"[] = ThisObject. "Total GainLoss"[]

ThisObject ^ "Farmers.Gain Loss"[] = 0 ["acre-feet"]

Note: Gain Loss in the accounting system is a volume.

Example:

Following is an example setting all of the account’s Slot Inflow to zero using a ForEach loop (note, this is just a sample method, this functionality can be accomplished much more easily using the compiled `Zero Slot Inflows` method):

```
FOREACH (STRING account IN AccountNamesByAccountType(ThisObject, “ALL”))
      ThisObject^(account CONCAT “.Slot Inflow”)[] = 0 [“cfs”]
ENDFOREACH
```

Note: *For more information on RPL and its use, see "RPL Sets" in RPLUserInterface.pdf.*

Execution

Object Level Accounting Methods provide flexibility in terms of when they can be executed, what they set and the data required. This section describes the execution times and some additional information on errors you may encounter during execution.

Execution Time

You are able to control when each Object Level Accounting Method is executed. Because Object Level Accounting Methods set the Slot Inflow and Gain Loss values on the accounts and these are required knowns for the account solution, this execution time allows you to control when the accounts solve. Depending on the application, you should execute the method as soon as possible once the information is known. For example, if you input Local Inflow, then you can configure your Slot Inflow method to execute at the Beginning of the run. Following are the possible execution times.

Execution Time	Description	Dependencies
Never	The method is never executed. This is only available for the default no-action method for each category.	None
Beg. of Run	At the beginning of the run, the method is executed once per timestep.	None
Beg. of Timestep Once	The method is executed once before each timestep's simulation.	None
Beg. of Timestep	The method is executed before each timestep's simulation and if dependent slots change. For this execution time, the method registers both simulation and accounting slots as dependencies. That is, if a value in a dependent slot changes, the method is put on the accounting queue to resolve. Note, this could have performance implications if slot dependencies cause methods to re-fire.	**Accounting** and **Simulation** Slots

Execution Time	Description	Dependencies
After Simulation	The method is executed after each timestep's simulation is complete and as accounting dependencies change. Note this is the default for compiled methods and the only execution time in releases prior to RiverWare 5.1. Technically, the method is executed as part of the “accounting beginning of timestep” which occurs after all rules have executed and all dispatching is complete. Each method registers accounting slot dependencies when executed. Therefore, if a value in a dependent accounting slot changes, then the method will re-execute.	**Accounting** Slots only

---

Note: *The run sequence including OLAM execution is described in “How the Accounting System Solves” on page 76.*

Data Requirements and Error conditions

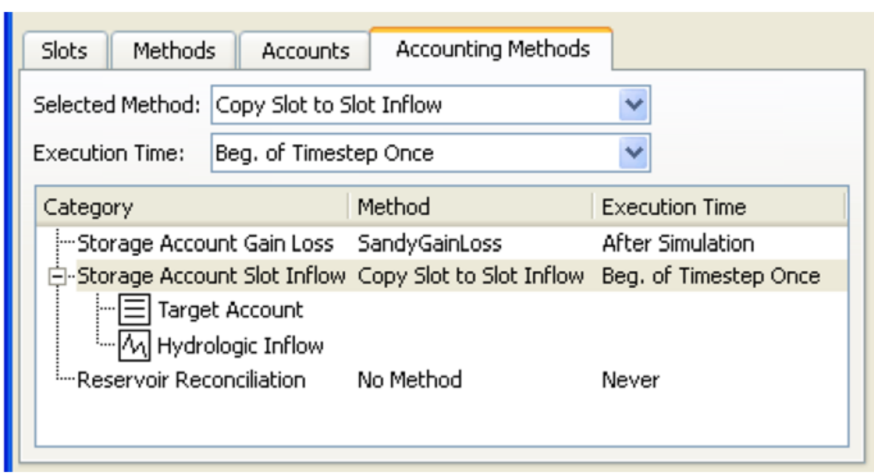
The Object Level Accounting Methods can only access information that exists at the time they are called or the method will terminate early. For example, the Object Level Accounting Methods set to execute “After Simulation” can only access accounting values at the previous timestep as accounts may not have solved yet. Usually, these methods can access physical values at the current timestep as the simulation has likely solved when “After Simulation” Object Level Accounting Methods are called. Input values can be accessed by any OLAM.

Additionally, if an OLAM attempts to set a slot’s timestep that has an Input value (I flag), the run will abort with an error: “Attempting to set an input value”. If you wish to input Slot Inflow / Gain Loss, then you will need to input Slot Inflow / Gain Loss on all accounts OR create a user defined method that does not attempt to assign to an input.

Selecting OLAMs and Execution Time

OLAMs and their Execution Time are selected from the object’s Accounting Methods tab.

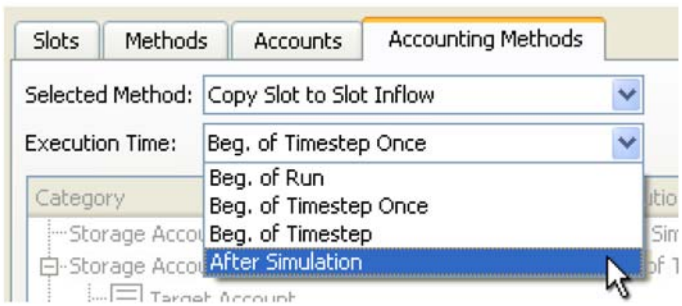
Figure 1-13:



The Accounting Methods tab works similar to the Methods tab. You select a category, then choose a method from the Selected Method: box at the top of the tab. An adjacent combo box labeled Execution Time: allows you to choose the Execution Time. The category / method list on this tab also contains the “Execution Time” column showing the currently chosen execution time for each selected method.

Additionally, you can select methods on multiple objects using the Multiple Object Method Selector described generally in "Multiple Object Method Selector" in ObjectDialogs.pdf. For OLAMs, see "Selecting Object Level Accounting Methods" in ObjectDialogs.pdf..

Figure 1-14:



Reconciliation

User defined accounting methods (and rules discussed later) can be used to ensure that the accounting system is reconciled with the physical system, i.e. the sum of the accounting releases is the same as the physical release and the sum of account storages is the same as the total object’s storage. RiverWare has no automatic checks to ensure that physical and paper water are reconciled. Each basin is unique and operates differently. As a result, it is the responsibility of the modeler to reconcile the two systems according to the legislation and operations of the basin. There are river basin that have legally decided that the accounting system can vary from the physical system on a daily basis and the total reconciliation happens on a monthly (or longer) timescale.

There are reconciliation categories on the aggregate diversion, reservoir, and water user objects. These categories can hold methods that set the Gain Loss slot, Slot Inflow slot, and/or accounting supplies. Reconciliation can also be accomplished using rules in a rulebased model. Further information on rules and accounting is given in “Accounting with Rules” on page 82.

How the Accounting System Solves

The accounting system does not dispatch in the same way that the physical water network simulation is dispatched, but there are some similarities. Account solution is triggered by the assignment of a value to a slot on the account. Assignment to account slots occurs in one of six ways: by user input, by a rule, by propagation through a supply, account solution, account level method, and by an object level accounting method on the object that references physical water quantities to introduce water into the accounting system (paper water). More information on each type of assignment is presented in the table “I/O Types” on page 13.

Accounts solve whenever sufficient known slot values are present to run a solution method. This is analogous to dispatching in the simulation network; however, in the case of the accounting system, this means that accounts may solve when a user edits the account values through the user interface, a rule sets a value, or a method sets a value. Therefore, a run is not required to make a single account solve.

Each time an account slot is given a value (during resetting of user input, execution of object level methods, rule execution, or during account solution), the account holding the slot is notified, and performs its own checks for over- and under-determination, and possibly solves at one or more timesteps. Some accounts solve “into the future” under some circumstances, even though the controller might be at a different timestep.

Outside of a run, accounts solve for as many timesteps as there is the required known data. This means that the account solution behaves similarly to a spreadsheet solution. If the user inputs an Outflow for each timestep in the accounting period for a storage object and all of the inflows, slot inflow and Gain Loss are known, then the account will solve for Storage at each timestep. This action is performed when the user hits the enter button. If the user then changes an Outflow value at the start of the accounting period, then the storage for the entire period will change because the solution equation for the storage account depends on the previous timestep’s Storage. By changing one Outflow, the Storage for the whole period is affected.

In a run, the solution is similar, but accounts will only solve into the future one timestep, even if enough information is given for the accounting system to solve for the entire run. For example, if Gain Loss and Slot Inflow of a storage account are known for an entire month starting on the 1st of the month, the storage will be solved for the entire month during that first timestep. During the course of the run, a transfer on the 10th of the month will reset the storage on the 10th of the month, and the 11th of the month, but won’t reset storage on the 12th or any subsequent date.

There are three controllers that allow accounting: Post-Simulation Accounting, Inline Simulation and Accounting, and Inline Rulebased Simulation and Accounting. These three controllers and their solution steps during a run are described in the following subsections.

Post-Simulation Accounting

The simulation must be run under the Simulation controller, then the post-simulation accounting process can be run under the Post-Simulation Accounting controller. The user must manually switch between controllers for each run. Accounts solve when they have the required knowns and resolve when a value in the solution equation changes.

Simulation Steps

The Simulation controller follows the procedure outlined in “Simulation Controller” on page 5

Accounting Steps

The Post-Simulation Accounting controller follows the procedure outlined below in the process of a run:

1. Initialize Accounting Run
    - Clear Iteration count
    - Clear Account States, i.e. Output values and Output flags
    - Set Accounting User Inputs
2. Accounting Beginning of Run
    - Execute Beginning of Run accounting methods for all objects.
    - Execute the Object Level Accounting Methods that are set to execute at `Beg. of Run`
3. For each timestep,
    - Execute the Object Level Accounting Methods that are set to execute at `Beg. of Timestep` and `Beg. of Timestep Once`.
    - Execute the Object Level Accounting Methods that are set to execute `After Simulation`.
    - If dependent accounting slot have changed, re-execute as necessary Object Level Accounting Methods that are set to execute at `Beg. of Timestep` or `After Simulation`.

Inline Simulation and Accounting

For each timestep, the physical simulation and accounting method execution are intermixed. Accounts solve when they have the required knowns and resolve when a value in the solution equation changes.

Simulation Steps

The simulation controller follows the procedure outlined below in the course of a run:

1. Initialization Simulation Run
    - Clear all output and values from previous runs for all timesteps in all series slots.
    - Set user inputs.
    - Propagate user inputs across links.
    - Determine first dispatch timestep. Click "Initialization" in Simulation.pdf. for more information.
2. Initialize Accounting Run
    - Clear Iteration count
    - Clear Account States, i.e. Outputs and Output flags
    - Set Accounting User Inputs
3. Execute rules in the Initialization Rules RPL set. Click “Initialization Rules Set” on page 50 for more information.

4. Simulation Beginning of Run
    - Execute Beginning of Run methods for all objects.
    - Evaluate Beginning of Run expression slots for all timesteps.
5. Accounting Beginning of Run
    - Execute Beginning of Run accounting method (account level methods) for all objects
    - Execute the Object Level Accounting Methods that are set to execute at `Beg. of Run`
6. For each timestep:
    - Set the controller clock to the timestep time.
    - Execute the Object Level Accounting Methods that are set to execute at `Beg. of Timestep` and `Beg. of Timestep Once`.
    - Execute Beginning of Timestep methods for all objects.
    - Evaluate Beginning of timestep, current timestep only expression slots
    - Dispatch objects until the queue is empty, simulating the effects of the user inputs and default values.
    - Execute End of Timestep methods on all objects.
    - Execute the Object Level Accounting Methods that are set to execute `After Simulation`.
    - For OLAMs set to execute `After Simulation` or `Beg. of Timestep`, re-execute if dependent slots have changed.
    - Evaluate end of timestep, current timestep only Expression slots
7. Execute End of Run Simulation methods on all objects.
8. Evaluate End of Run expression slots.

Inline Rule-based Simulation and Accounting

For each timestep, the rule-based simulation is run where slots are set by rules, the simulation objects dispatch, and user defined accounting methods execute. Remember that accounts solve when they have the required knowns and re-solve when a value in the solution equation changes. This can happen at any time throughout the course of the run or even outside of the run. This section describes the inline rulebased simulation and accounting steps in more detail. First, the run steps are presented in paragraph format, then they are presented in bullet format.

Rulebased simulation is described at: "How Rulebased Simulation Works" in RulebasedSimulation.pdf. As a reminder, execution of rules and dispatching (simulation) of objects are inter-mixed and may be mutually-dependent. At each timestep, the controller cycles the following three steps:

1. Processing the object dispatch queue. Objects check their dispatch conditions and may try to dispatch (or re-dispatch) after a “dispatch slot” on the object changes value. An object maintains a list of its dispatch slots and dispatching only occurs if it has sufficient known values.
2. Processing the accounting queue. Object Level Accounting Methods with the execution time of `Beg. of Timestep` can have simulation slot dependencies so that when the dependent simulation slot changes, the method is placed onto the accounting queue for processing. The accounting queue is processed in a first in, first out order.

3. Executing rules from the agenda one at a time. When a rule is successful, all its slot assignments are made. If it is not successful, none of its assignments are made, i.e. never are some, but not all of the assignments made. When the rule succeeds, the slot cell whose value was changed is given the priority of the rule and the “R” flag. These values show up in the rules analysis dialogues. Each rule maintains a list of the slots on which the outcome of the rule is “dependent”, namely those slots that were read during the rule execution. A rule will re-execute after one or more of its dependency slots changes value. Thus, objects that are dispatching can cause rules to re-execute by changing the values of slots upon which the rules depend, while rules that change dispatch slot values can cause simulation objects to dispatch. In the accounting system, a rule can change an account slot's value, which may cause the account to solve and may propagate changes throughout the accounting system. This can happen during a rule execution, but not during an object dispatch. A change in a simulation slot can also cause an Object Level Accounting Method to go onto the accounting queue if the slot is a dependent simulation slot of the method,. This can occur during rule execution or during an object dispatch,

At the start of the timestep (after beginning of timestep behavior has occurred where object level accounting methods may be executed), the dispatch queue (step 1) is fully processed until empty. This simulates the effects of user inputs. In this processing of the dispatch queue, objects may dispatch one or more times or may not dispatch at all depending on the dispatch slots that are set or changed throughout the course of the dispatching. Then the accounting queue (step 2) is processed to execute any Object Level Accounting Methods that have come onto the queue.

Once this queue is empty, the rules controller moves to the third step and starts with the full set of enabled rules on its “agenda” in priority order. This ensures that each enabled rule gets at least one chance to execute. The controller tries to execute the first rule on the agenda, in user specified order, at which time the controller removes the rule from its agenda. A successful rule may put objects on the simulation dispatch queue or Object Level Accounting Methods on the accounting queue. After a rule succeeds and sets a value, the controller returns to step 1 and dispatches all objects on the dispatch queue and then in step 2 executes all Object Level Accounting Methods on the accounting queue.

When the dispatch and accounting queues are again empty, the controller returns to step 3 and executes the next rule on its agenda. Once a rule succeeds, it returns to step one and processes the dispatch queue and then the accounting queue. It continues cycling until the dispatch and accounting queues and the rules agenda are empty.

When all simulation dispatching is complete and the rules agenda is empty, Object Level Accounting Methods may execute under the accounting controller and then the simulation moves to the next timestep.

Rule-based Simulation Steps

The rulebased simulation controller follows the procedure outlined below:

1. Initialization Rulebased Simulation Beginning of Run
    - Clear all output and values set by rules from previous runs for all timesteps in all series slots.
    - Set the controller priority to 0.
    - Set user inputs.
    - Propagate user inputs across links.
    - Determine first dispatch timestep. For more information, see: "Initialization" in Simulation.pdf.
2. Initialize Accounting Run

- Clear Iteration count
- Clear Account States, i.e. Outputs and Output flags
- Set Accounting User Inputs

3. Execute rules in the Initialization Rules RPL set. For more information, see:"Initialization Rules Set" in RPLUserInterface.pdf.
4. Rulebased Simulation Beginning of Run
   - Execute Beginning of Run methods for all objects.
   - Evaluate Beginning of Run expression slots for all timesteps.
5. Accounting Beginning of Run
   - Execute Beginning of Run accounting method for all objects
   - Execute the Object Level Accounting Methods that are set to execute at `Beg. of Run`
6. For each timestep:
   a. Set the controller clock to the timestep time.
   b. Execute the Object Level Accounting Methods that are set to execute at `Beg. of Timestep` and `Beg. of Timestep Once`.
   c. Set controller priority to 0.
   d. Execute Beginning of Timestep methods for all objects.
   e. Evaluate expression slots that have evaluate-at-beginning-of-timestep selected
   f. Put all rules on the agenda (in priority order - whether 3,2,1 or 1,2,3 is user-selectable)
   g. Do the following processes until the dispatch queue, the accounting queue, and the rules agenda are empty
      - Process 1 - Process the Dispatch Queue: Dispatch objects (as in basic Simulation) until the queue is empty, simulating the effects of any user inputs and default values and any recent changes to slots by rules. For each slot changed by this dispatch:
      1. Put all rules that depend on the changed slot on the agenda, if it's not already therelfjslfjslfjslfkjslfjslkls.
      2. If the slot is a dispatch slot, check dispatch conditions and if necessary put the object containing the slot on the simulation dispatch queue.
      3. Put all Object Accounting Methods with execution time of `Beg. of Timestep` that depend on the slot on the accounting queue.
      4. If the slot is an accounting slot, notify the account that it may need to solve.
      5. Once the Dispatch Queue is empty, move on to process 2.
      - Process 2 - Process the Accounting Queue. Execute any Object Level Accounting Methods that have come onto the accounting queue. For each slot changed as a result of method execution:
      1. If the slot is an accounting slot, notify the account that it may need to solve.

2. If the slot is an accounting slot, put all Object Level Accounting Methods that have a dependency of this accounting slot onto the accounting queue for execution if the method's execution time is also Beg. of Timestep.
3. Once the Accounting Queue is empty, move on to process 3.

- Process 3 - Execute a Rule on the agenda: Set the controller priority to the priority of the next rule on the agenda (either in order 3,2,1 or 1,2,3 based on user-selection), and fire this rule. If not successful, continue firing one rule at a time until a rule is successful and at least one slot is set in the model. Each rule is removed from the agenda after it fires. Once a rule is successful:

1. Apply the slot changes, giving the changed slot cell the priority of the controller (i.e. the last rule that fired)
2. Add to this rule's dependency's list each slot that was read by this rule.
3. For each slot set by this rule, put all rules that depend on the changed slot on the agenda if the rule is not already there.
4. For each slot changed by this rule, if the slot is a dispatch slot, check dispatch conditions and if necessary, place the object containing the slot on the simulation dispatch queue.
5. For each slot changed by this rule, if the slot is an accounting slot, notify the account that it may need to solve.
6. For each slot changed by this rule, if the slot is a dependent simulation slot for an Object Level Accounting Method with an execution time of Beg. of Timestep, add the method to the accounting queue.

h. Return to Process 1 and repeat until the Agenda and the Queues are empty.
i. Set controller priority to zero, and execute End of Timestep methods on all objects.
j. Execute the Object Level Accounting Methods that are set to execute After Simulation.
k. If dependent accounting slot have changed, re-execute as necessary Object Level Accounting Methods that are set to execute at Beg. of Timestep or After Simulation.
l. Evaluate end of timestep, current timestep only Expression slots

7. If the number of run cycles performed is less than the number of run cycles specified, return to item 6 and loop through each timestep again. For more information, see "Run Cycles" in RulebasedSimulation.pdf.
8. Execute End of Run Simulation methods on all objects.
9. Evaluate End of run expression slots.

Accounting with Rules

In an Inline Rulebased Simulation and Accounting run, rules can be used for two purposes: setting slots on the physical system and setting supplies in the accounting system. In the accounting system, rules are used to move water between two accounts by setting Supplies. To clarify the two types of assignments, we will define simulation rules/assignments as follows:

> Simulation or Physical Rules/assignments
>
> Rules and/or assignments that set values in the physical simulation system. For example, these rules set reservoir Outflow, Storage, or Pool Elevation.

We will also define accounting rules/assignments as follows:

> Accounting Rules/assignments
>
> Rules and/or assignments that set slots or supplies in the accounting system

Both simulation rules, i.e. rules that set outflows and storages, and accounting rules, i.e. rules that set supplies, can be part of the same ruleset and even assignments within the same rule. Accounting rules may be interspersed with physical rules that set simulation slots. Or, all of the Simulation rules can execute, then all of the accounting rules can execute. The structure of the ruleset depends on the application. In addition, accounting assignments and simulation assignments can be within the same rule. For example, when a physical release is made, the release can be allocated to the appropriate supplies in the same rule. Following is a description of the possible ways an accounting model can interact with the physical simulation:

After-the-Fact Accounting:

The physical simulation is run completely without any rule effects, i.e. all objects dispatch because of input data. Rules are used to determine how the water already released in the physical simulation should be classified in the accounting system. This type of application has only Accounting Rules, all of the physical simulation is the result of input data. Typically this type of system is used in an operations mode to determine how yesterday’s final releases should be classified. For example, if Reservoir.Outflow = 100cfs yesterday. In the accounting system, 20cfs of that was for fish minimums and 80cfs was for downstream demands.

Physical and Accounting (Concurrent on each timestep):

For each simulation assignment/rule, there is an accounting assignment/rule. In this application, for example, each physical release corresponds to a release in the accounting system. The physical operations often depend on the results of the accounting operation. This is usually the most intuitive to describe but is often difficult to implement. Each time a release is increased or decreased, the accounting system must also be increased or decreased by the same amount. If there are physical constraints that limit the amount of water that can be released, it quickly becomes confusing as to what supply needs to be set. For example, Rule 2 sets Reservoir.Outflow = 80 cfs and in the accounting system sets that 80cfs was for downstream demands. Rule 1 then sets Reservoir.Outflow = 100cfs and in the accounting system sets that the additional 20cfs was for minimum flows.

Physical then Accounting (Consecutive per timestep)

For each timestep, first all of the simulation rules execute and set releases. Then, accounting rules execute and determine how those releases should be classified. This is similar to after-the-fact accounting but rules are used to determine the physical releases as well. This type of system is convenient because the physical rules have access to account values at the previous timestep and release constraints can be more easily incorporated. The accounting rules look at the total release and use logic to determine the accounting releases. For example, Rule 3 sets Reservoir.Outflow equal to 80cfs, Rule 2 sets Reservoir.Outflow equal to 100cfs. Then in the accounting system, Rule 1 sets that of the 100cfs released from the Reservoir, 20 cfs was for minimum flows and 80 cfs was for downstream demands.

Water Rights:

Accounting rules allocate flow to the accounts, then physical rules look at the results to drive the simulation. This is covered in the Water Rights Allocation section of the help. For more information, see Chapter 5, “Water Right Allocation,” on page 167.

Following are considerations that should be followed when using rules with accounting.

Setting Slots versus Setting Supplies

In the physical system, rules are used to set values on slots such as Reservoir.Outflow or WaterUser.Diversion Requested. When making RPL assignments in the accounting system, the user has to be careful to about which values to set. Rules can be used to either set an accounting slot or a supply in the accounting system. Setting of slots can only be accomplished if there is no supply connecting the slot to another slot. If there is one or more supplies connecting the slot, then the rule must set the supply. Lets consider the following example:

> Example
>
> A storage account on a reservoir has the typical slots including inflow, outflow and storage. If there is no supply (actually a demand) leading out of the account, then a rule can be used to directly set the outflow: “Reservoir^Account.Outflow”. If there is one or more supply leading out of the account, then it is not possible to directly set the outflow. Instead one of the supplies should be set by the rule such as “Reservoir Acct to Reach Acct.Supply []”. Setting of supplies is described in the following section.

Typically, the user should use rules to set supplies, not accounting slots. There are a few cases where the user must set the slot directly. For example, on a diversion account, if the user wishes to specify the Depletion using a rule, then the left hand side of the rule would have the following format: WaterUser^DiversionAccount.Depletion[].

Setting Supplies

For accounting assignments, the left-hand side of the assignment statement will consist of a Supply name with the following syntax:

```
“SupplyName.Supply”[] =
```

For this syntax, the “SupplyName.Supply” is a String. The “.Supply” portion of the string is necessary to tell RiverWare that this is a supply in the Accounting system and not a slot in the physical system. The SupplyName.Supply either can be typed by the user or selected from the palette using the supply option in the selector dialog.

Because accounts will not solve until the User Defined Accounting Methods have executed and set the Gain Loss and Slot Inflow, the accounting rules can typically reference current timestep information on the simulation objects but can only reference previous timestep information on the accounting system. Within this limitation, simulation rules can also reference values in the accounting system.

Example:

A water district and a power company share ownership of the water in a reservoir. On any given day, the water district makes calls to release their water to meet downstream demands. The power company releases water to meet power demands. A simple rule for this reservoir may look like:

Reservoir.Outflow[] = FarmerDemand() + PowerDemands()

WaterDistrictAccount.Supply[] = Farmer Demand()

PowerAccount.Supply[] = Power Demands()

In this rule we have set both the Outflow to the reservoir in the physical system and then classified that release in the accounting system. The functions FarmerDemand() and Power Demands() can reference the available storage in each account at the previous timestep and reference the current demand:

FarmerDemand() -> Min (FarmersData.DiversionSchedule[],

PreviousAccoutStorage("WaterDistrict"))

PowerDemand() -> Min (PowerData.Demand[],

PreviousAccoutStorage("Power"))

PreviousAccountStorage(STRING Account) -> VolumeToFlow (Reservoir ^ Account.Storage[@"Previous Timestep"], @"Current Timestep")

Using this approach, the amount of water that either user could release is limited by the volume of water stored in their account at the previous timestep.

Reconciliation with Rules

Rules can be used to ensure that the physical system is reconciled with the accounting system, i.e. the total physical release is the same as the total accounting release. RiverWare has no automatic checks to ensure that physical and paper water are reconciled. Each basin is unique and operates differently. As a result, it is the responsibility of the modeler to reconcile the two systems according to the legislation and operations of the basin. There are river basin that have legally decided that the accounting system can vary from the physical system on a daily basis and the total reconciliation happens on a monthly (or longer) timescale.

Predefined RPL Functions for Accounting

There are predefined RPL functions that deal specifically with accounts, supplies, and exchanges in the accounting system. Below is a brief description of some accounting specific predefined functions. More information can be found in "Abs" in RPLPredefinedFunctions.pdf.

AccountNamesByAccountType

This function returns a list of names of Accounts on a specified Object having the indicated Account type, sorted in ascending Account priority date order. Accounts which don't have a priority date are at the end of the list, sorted in ascending name order.

AccountNamesByWaterOwner

This function returns a list of names of Accounts on a specified Object having the indicated WaterOwner, sorted in ascending Account priority date order. Accounts which don't have a priority date are at the end of the list, sorted in ascending name order.

AccountNamesByWaterType

This function returns a list of names of Accounts on a specified Object having the indicated WaterType, sorted in ascending Account priority date order. Accounts which don't have a priority date are at the end of the list, sorted in ascending name order.

AccountNameFromPriorityDate

This function evaluates to the name of the Account having the specified priority date.

AccountNamesFromObjReleaseDestination

This function returns a list of names of Accounts on a specified Object where the attributes of the outflow Supplies of the Accounts match the given ReleaseType and Destination. The list is sorted in ascending Account priority date order; Accounts which don't have a priority date are at the end of the list, sorted in ascending name order.

AccountPriorityDate

This function evaluates to the priority date of the Account, on the specified Object, having the specified name.

Destinations

This function evaluates to a list of user defined Destinations.

DestinationsFromObjectReleaseType

This function returns a list of unique names of Destinations of Supplies which represent outflows from a specified Object, and which have the indicated Release Type.

GetAccountFromSlot

Given a Slot, the function return its parent account name as a String. It is an error if the slot is not on an account.

GetObjectDebt

This function evaluates to the sum of the debts to all accounting exchanges which may be paid by supplies on the given object. If there are no exchange paybacks on the given object, the debt is zero.

GetPaybackDebt

This function evaluates to the value of the debt slot of the given exchange payback source at the given timestep.

ObjAcctSupplyByWaterTypeRelTypeDestType

This function returns a list of objects, accounts and supplies that match the given arguments. It returns a list of triplets{OBJECT object, STRING account, STRING supply}, where the object^account is served by the supply, and the object is in the given subbasin, the supply has the given release type and destination type, and the supplying account (upstream end of the supply in the returned triplet) has the given water type.

ObjectsFromAccountName

This function returns a list of the objects that contain an account with the given name and account type.

ObjectsFromWaterType

This function returns a list of the objects that have an account with given water type and account type.

ReleaseTypes

This function returns a list of the names of all user defined ReleaseTypes in the Water Accounting System Configuration.

ReleaseTypesFromObject

This function returns a list of unique names of ReleaseTypes of Supplies which represent outflows from a specified Object.

SolveWaterRights and SolveWaterRightsWithLags

This water accounting function invokes the Water Rights Allocation method on a computational subbasin. For more information, see Chapter 5, “Water Right Allocation,” on page 167.

SumAccountSlotsByWaterType

This function sums the values of all accounting slots of a given name on accounts of a given water type

SupplyNamesFrom, SupplyNamesFrom1to1

This function returns a list of names of Supplies which represent outflows from given Accounts and have the indicated ReleaseType and Destination.

SupplySlotsFrom, SupplySlotsFrom1to1

This function returns a list of Supply slots of Supplies which represent outflows from given Accounts and which have the indicated ReleaseType and Destination.

SupplyNamesFromIntra, SupplyNamesFromIntra1to1

This function returns a list of names of Supplies which represent internal flows from given Accounts and which have the indicated ReleaseType and Destination.

SupplySlotsFromIntra, SupplySlotsFromIntra1to1

This function returns a list of Supply slots of Supplies which represent internal flows from given Accounts and which have the indicated ReleaseType and Destination.

SupplyNamesTo, SupplyNamesTo1to1

This function returns a list of names of Supplies which represent inflows to given Accounts and which have the indicated ReleaseType and Destination.

SupplySlotsTo, SupplySlotsTo1to1

This function returns a list of Supply slots of Supplies which represent inflows to given Accounts and which have the indicated ReleaseType and Destination.

SupplyNamesToIntra, SupplyNamesToIntra1to1

This function returns a list of names of Supplies which represent internal flows to given Accounts and which have the indicated ReleaseType and Destination.

SupplySlotsToIntra, SupplySlotsToIntra1to1

This function returns a list of Supply slots of Supplies which represent internal flows to given Accounts and which have the indicated ReleaseType and Destination.

WaterOwners

This function returns a list of the names of all WaterOwners defined in the Water Accounting System Configuration.

WaterTypes

This function returns a list of the names of all WaterTypes defined in the Water Accounting System Configuration.

Chapter 2
Accounting User Interface

This document describes the user interface for the Accounting system.

This chapter covers the following topics:

- "Enabling Accounting" on page 90.
- "Synchronizing Accounting" on page 91.
- "Account System Configuration" on page 92.

Enabling Accounting

To enable an accounting model, first accounting has to be enabled from the menu.

- Open the Accounting menu on the tool bar and click Enable Accounting option. A check mark is added next to the menu selection. This also enables the remaining options in the Accounting menu.

Second, one of the three accounting controllers has to be selected.

- Open the Run Control dialog and select either Inline Simulation and Accounting, Post-Simulation Accounting, or Inline Rulebased Simulation and Accounting.

Synchronizing Accounting

The accounting system must have the same timestep as the physical model. By default, it should have the same timestep, but there are steps the user could take to create a discontinuity. To synchronize accounts with the run control:

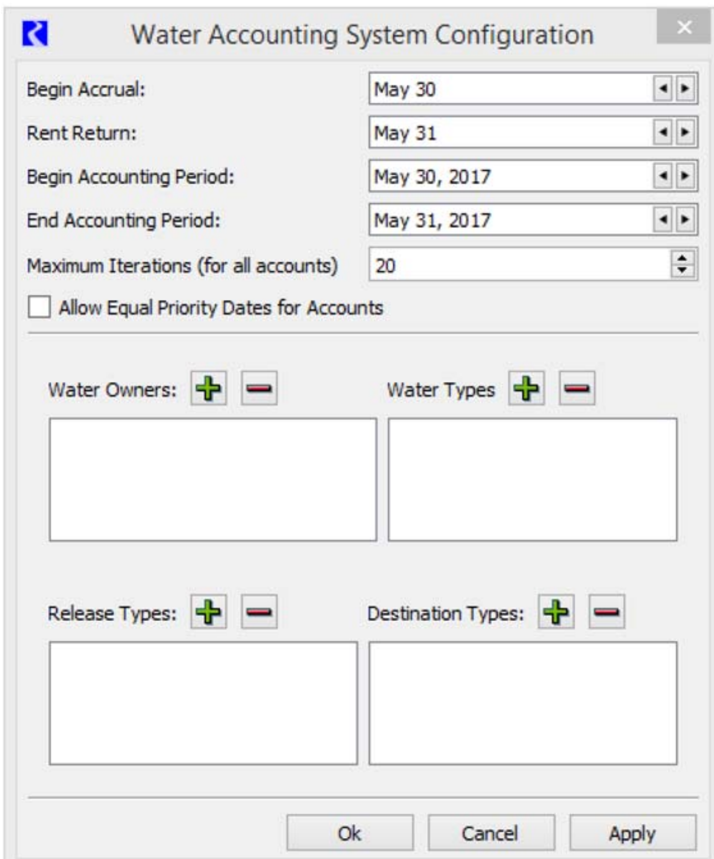
- Click Workspace > Objects > Synchronize Objects.
- In the dialog, select the toggle to Include Accounting Objects, and then click Sync. Then confirm by clicking OK in the Confirmation dialog.

Account System Configuration

The accounting system is configured separately from the simulation system. In the Water Accounting System Configuration dialog, the user is able to set values that affect the entire accounting system like time range, water type, and water owner definitions.

- Open the Water Accounting System Configuration dialog by selecting Accounting > Account System Configuration on the tool bar.

Figure 2-1: Water Accounting System Configuration



The Accounting System Configuration contains the following:

Begin Accrual

A Begin Accrual Date is used by storage accounts. On this date, the accrual in each storage and diversion account is reset to zero. In addition, this date is used by the carryover methods on the account.

Rent Return

A rent return date can be configured for the entire accounting system. Currently, this value is not used anywhere.

Begin Accounting Period

An accounting run, like a simulation run, has a period of dates for which accounts solve. The `Begin Accounting Period` is where the users sets the time range of the accounting system. The date can be before the initial simulation timestep but cannot be after the initial simulation timestep.

End Accounting Period

The End Accounting Period is where the user sets the timestep where the accounting system ends. The end timestep can be after the last simulation timestep but cannot be before it. In other words, the accounting system must cover the entire simulation time but can extend beyond it in either direction.

As a model evolves, you might change the run times (start, finish, timestep size) with the run control dialog (`Control` > `Run Control Panel...`). Objects created before the changed run times will differ from objects created after the changed run times. Similarly, accounts created before the run time change will differ from those create after the run-time change with respect to their slots' time ranges and timestep sizes. RiverWare copies the run time information from the run controller to the accounting system so that it knows if there is a possibility that accounts are not synchronized with the run control information.

Max Iterations

As with the physical system, slots in the accounting system can be set many times in a run. The accounting system configuration dialog box allows you to put a limit on the number of times that any slot can be set; if that limit is exceeded for any slot, the run aborts with an error message. Be careful to set this limit high enough for the solve-into-the-future behavior of some accounts.

Below the Max Iterations are areas of the dialog that can be used to define Water Owners, Water Types, Release Types and Destination Types. Each one can be added by clicking on the  icon or deleted using the  icon.

Allow Equal Priority Dates for Accounts

This toggle controls whether the accounting system will allow accounts to have priority dates that are the same, or if each account's priority date must be unique. Priority dates are used to allocate water in rules functions.

Water Owners

Each account can have a water owner which is used to classify the account. The Water Owners area allows the user to create the set of water owners available to the accounts.

Water Types

Each account can have a water type associated with it. The Water Types area allows the user to create the set of water types available to accounts.

Release Types

Release Types are used to classify supplies. The Release Types area allows the user to create the set of release types available to supplies. For example, the user can specify that there are Fish and Irrigation Release types. The release type applies to a supply, this is the water coming out of or going into an account. We will not add release types at this time.

Destination Types

Destination Types are also used to classify supplies. The Destination Types area allows the user to create a set of Destination Types available to supplies. Both Destination Types and Release types become very useful when using the rule controller to define how much and from where water is released.

Note: Units: *Prior to version 6.3, you configured accounting specific units in the Accounting System Configuration. Now units are specified in the active Unit Scheme as described "Unit Schemes" in Units.pdf.More information that is specific to accounting units can be found in “Slots” on page 101.*

Note: Supply Name Format: *Although not specified in this dialog, the supply name format can be configured; this is another system configuration that you may wish to do before building a model. The format for supply names can be controlled by the user in the Supply Name Format dialog. For more information, see “Supply Name Format” on page 129. The format specified in this dialog is applied to new supplies and can be applied to existing supplies from the Supply Manager.*

The Accounting Network

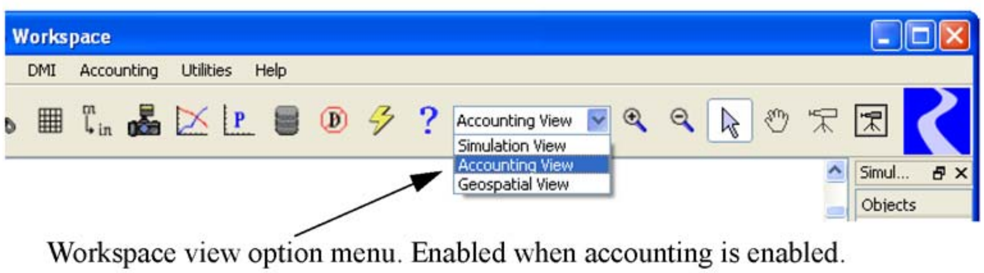
The RiverWare workspace shows the layout of the physical system of your model. For more information on the workspace, see "Introduction" in Workspace.pdf. In an accounting model, there is secondary workspace called the Accounting View which shows the accounting network in the model. The Accounting View provides a graphical layout of the Accounting network including objects, accounts, links, and supplies. Note that, if objects with accounts and supplies are exported/imported, the supplies and accounts can also be exported/imported. For more information on the options available to import or export objects with their accounts and supplies, see "Importing and Exporting Objects" in Workspace.pdf.

The following section describes how to access the accounting view, describes features of the Accounting View, workspace options, display properties, and graphical account aggregation.

Accessing the Accounting View

If accounting is enabled, the accounting network is displayed in the `Accounting View`, and can be selected by the option menu on the toolbar as shown in the following screenshot.

Figure 2-2: Accessing the Accounting view



The accounting view provides a representation of the physical objects, as well as, the accounts on those objects. The first time the user switches from the Simulation View to the Accounting View, RiverWare uses the location of the objects to create a default layout in the accounting view. This default layout is similar to the Simulation View but typically, the user must rearrange objects and accounts to get a realistic and usable layout.

- Click on the Workspace view option menu shown above and switch to the Accounting View.

Features of the Accounting View of the Workspace

Accounts in the accounting view are displayed in the *object territory* of its parent simulation object, providing a spatial association of the account with its simulation object. An object territory is a rectangular box with the object icon and name in the upper left corner. Icons representing the accounts on that simulation object are displayed within the territory in a single row. The position of the accounts in the territory are configured through a drag and drop operation; however, RiverWare provides an initial default ordering.

Figure 2-3:

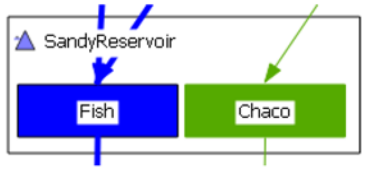
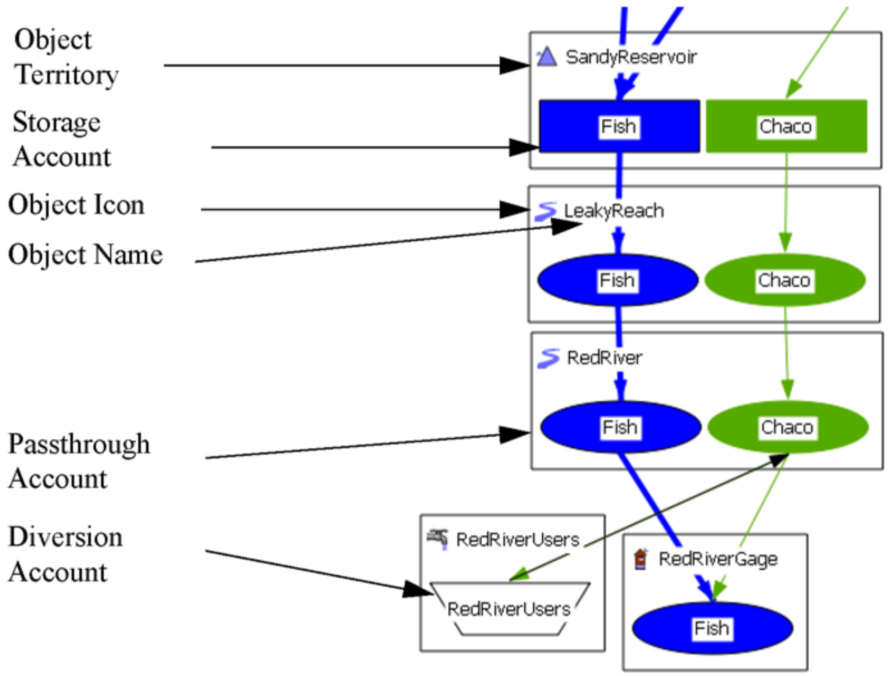


Figure 2-4:



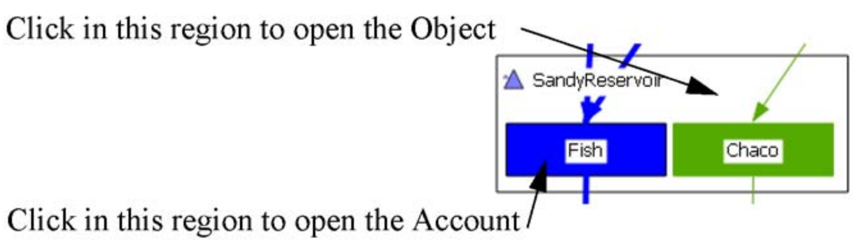
The accounting view can represent the four account types: storage accounts, passthrough accounts, diversion accounts, and instream flow accounts. Storage accounts are represented by rectangular icons, pass through accounts are represented by elliptical icons, diversion accounts are represented by an isosceles trapezoid, and instream flow accounts are represented by a pointed rectangle. The account name is presented within its icon; however, if the account name is too long to fit within its account icon, the name is truncated and marked with an *****. Supplies are represented by a line between two accounts with the directional supply-demand relationship represented by an arrowhead on the demand-side of the supply. Supplies between accounts on the same object (transfers) are represented by a curved line.

The position of the object territories is configured through a drag drop operation just as icons can be moved in the simulation view. The first time the accounting view is activated for a model, RiverWare will attempt to position the accounts in a meaningful way; however, some manual tuning will be necessary particularly in areas of divergence or convergence. The user can force the network to layout again by right clicking with the mouse on the workspace, and selecting `Layout Network`. The positions of the objects in the simulation and accounting views are not coupled and can be move independently.

Accounting View Navigation Techniques

The Accounting View workspace can be used for model building and navigation similar to the main Simulation View workspace. This includes the construction of simulation objects and links, not just accounts and supplies. The user can add objects, create accounts on objects, create links and supplies, etc. Context sensitive menus are activated by the right mouse button. In the object territory, right clicking gives the user options to create links, open the object, and add accounts among other features. Right clicking on an account gives the user the option to open the account, configure the account and add supplies to the account.

Figure 2-5:

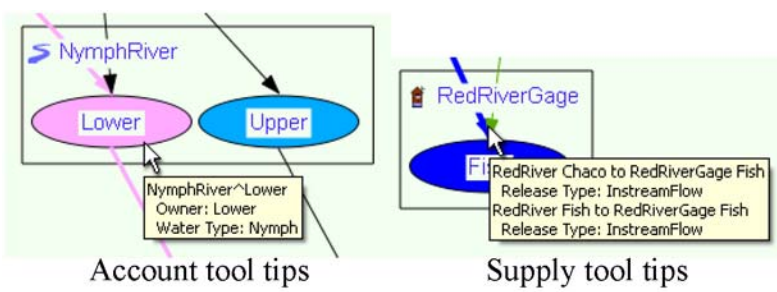


Tool Tips in the Accounting View

Mousing over a component in the accounting view will provide the name of that component as a tool-tip. Place the mouse on the component and leave it there until the name appears, after approximately one second.

Information shown includes the name and the following user-defined attributes for accounts and supplies:

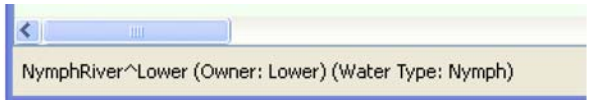
Figure 2-6:



- Account attribute: Water Owner
- Account attribute: Water Type
- Supply attribute: Release Type
- Supply attribute: Destination Type

Also the information is shown in the workspace status bar (lower left corner). This can be useful because as long as the mouse is over the item, the name is shown in the status bar. Tool tips only show the name for a few second.The status line includes all relevant attributes, regardless of whether or not those attributes are defined in the model file (edited in the Accounting System Configuration dialog). For attributes without definitions, a value of “NONE” is shown, e.g. (“Destination Type: NONE”).

Figure 2-7:



Display Properties

The appearance of links, objects, accounts, and supplies on the workspace can be manipulated using the Display Group Editor. The dialog is accessed by:

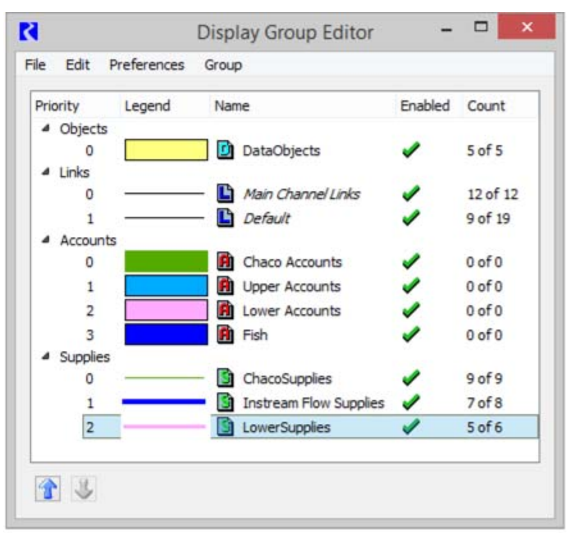
- Selecting Workspace > Display Group Properties

or

- Right-clicking with the mouse on the workspace. This will bring up a window with the option to select Display Group Properties.

The Display Group Editor provides the ability to create groups of links, groups of objects, and if accounting is enabled, groups of supplies and groups of accounts. The display properties (including color, line width, shading, etc.) and the membership of a group can be changed by double-clicking on that group. The membership of the groups need not be mutually exclusive. The groups are prioritized. For example, if two link groups contain the same link, the group highest in the list (lowest priority number) will be used to set the display properties of that link. Display groups can be enabled and disabled by clicking on the check mark in the *Enabled* column.

Figure 2-8:

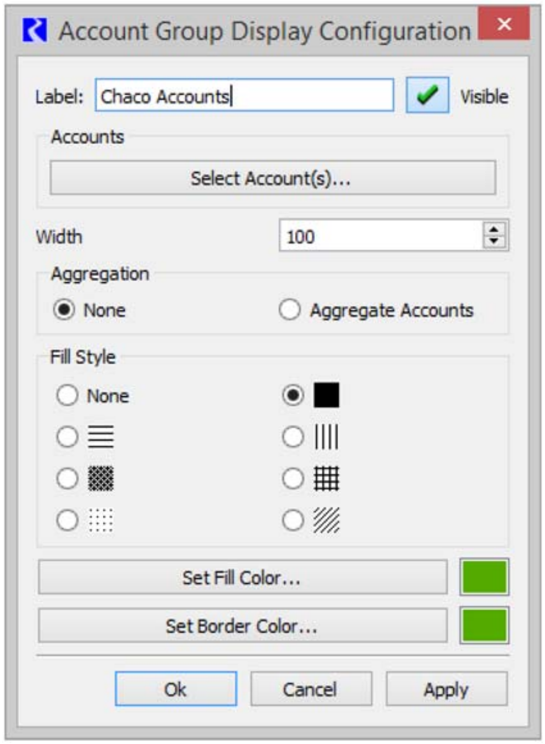


- Right-click on the workspace and select Display Group Properties. The following dialog opens:

Account Groups

Add a new Account Group through the Group -> Add Account Group. Double-clicking on an account group entry will bring up the Account Display Group Configuration dialog.

Figure 2-9:



The membership of the group and the fill style, fill color, account width, account aggregation, and border color of the group members can be set with this dialog. The membership of Accounts within the group is controlled by pressing the `Select Account(s)` button. The selection uses the standard Object/Slot/Account/ selector dialog and wild cards and filters can be used to select the appropriate accounts. An account can also be added to an Account Display Group by right-clicking on the account on the workspace, mousing over `Add to Account Group`, and clicking on the desired Account Display Group. If the account is already a member of an Account Display Group, this option will be replaced with `Account Group Membership`. Mousing over this option will show the display group to which the account belongs.

Supply Groups

Similar to the Account groups, Supply groups are used to control the appearance of supplies or accounting links.

Properties

The display properties of the accounts and supplies are configured through the display group editor. The display group editor allows users to define groups of objects and the display properties on those groups. The groups are prioritized, allowing an item to span multiple groups. The properties of the highest priority group containing an item will be used to draw the item. Configurable account display properties include:

- Border color - the color used to draw the outline of the account’s geometric representation.
- Fill Color - the color used to fill the area within the account’s geometric representation.
- Fill Style- the pattern used to fill the area with the account’s geometric representation (e.g., solid vs. a

crosshatch pattern).

- Width - the width of the account's geometric representation. The geometric representation will stretch to accommodate the account's name up to this limit, at which point, the account name will be truncated. Hovering the mouse over the account will provide the full name.
- Visibility - controls whether or not the account will be shown in the network.
- Aggregation - a built in ability to aggregate the visual representation of one or more accounts.
- Configurable supply display properties include:
- Line Color - the color used to draw the supply.
- Line Style - the style used to draw the supply (e.g., solid line vs. dashed line).
- Line Width - the width of the line.
- Visibility- controls whether or not the supply will be shown in the network

The user can also select to show the simulation links (e.g. Reservoir.Outflow to Reach.Inflow) on the Accounting View. This option is selected from the Display Group Editor dialog `Preferences` > `Show Links in Accounting View`. Using link groups in the Display Group Editor, the links can be colored and the width can be changed to distinguish them from the accounting supplies. Showing the simulation links helps when building an accounting model to show which objects are connected to which other objects. This is useful because accounting supplies can only connect objects which are physically linked in the accounting system.

Graphical Account Aggregation

The account aggregation capability of the account display group provides functionality to group similar accounts to avoid clutter on the workspace. When accounts are aggregated, all the accounts in the aggregation will be represented by a single icon that looks like a "stack" of two accounts icons. This representation is labeled with the name of the display group that aggregated the accounts. A tooltip (i.e. mouse over) provides the names of all the accounts in the aggregation.

Configuring Accounts through the Open Account dialog

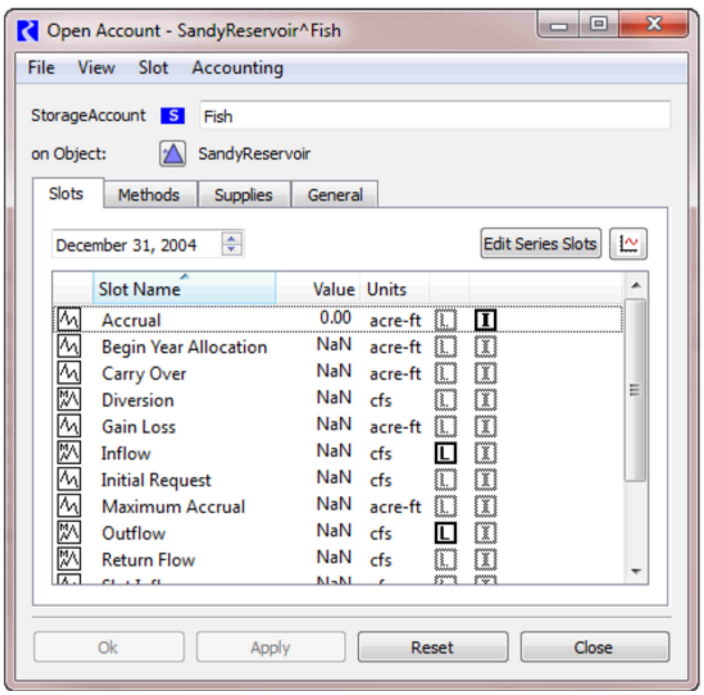
The Water Accounting System Configuration dialog is used to configure all new or existing accounts at one time. To configure an account individually, instead use the Open Account dialog. The Open Account dialog can be used to both configure an individual account and view the data associated with that Account.

To access the Open Account dialog, use one of the following methods:

- From the object dialog, switch to the Accounts tab. Select the account and then select Account > Open Account... from the menu bar or right click on the account and select Open Account... from the context menu.
- From the Edit Account dialog, select View > Open Account.
- From the Accounting View of the workspace, right click on the account and select Open Account... from the context menu

The Open Account dialog has the following tabs: Slots, Methods, Supplies, and General, as shown in the following screenshot.

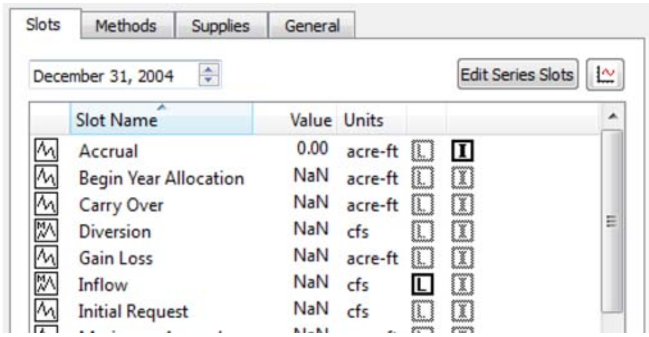
Figure 2-10:



Slots

The Slots tab shows all of the slots on the account. Double click on a slot to bring up a view of that slots data. If the slot is a multi-slot, both the total and the components will be shown. This is another location where accounting data can be entered or edited. In the Slots tab, there is also a date time spinner to scroll to a specific date and a plot button to plot the selected slot(s).

Figure 2-11:



From the Slots tab, you can double click any slot to see the values in a standard slot dialog. From the slot dialog, you can view variables shown in the following table. You can make changes to units, scale, precision and format which actually result in a Slot Exception in the Unit Scheme as described in "Unit Schemes" in Units.pdf.

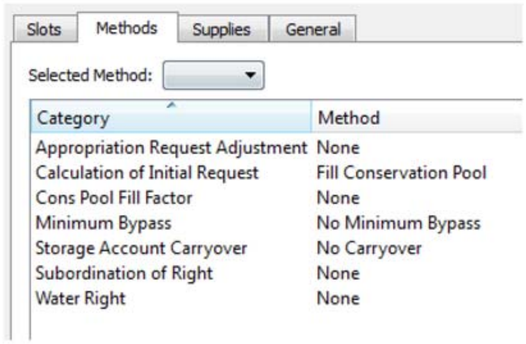
Note: *The Open Slot configuration dialog is the only location to set the accounting slot's* Convergence.

Editable Variable	Description
Units	The unit displayed in the user interface
Scale	A multiplier of the unit (e.g., 1 cms or 10 cms), so its effect is to be a divisor of the value displayed.
Precision	The number of digits after the decimal point that are displayed
Format	A choice amongst float and scientific. Float means floating point, displayed with the precision number of digits after the decimal point, e.g., 55.99, if precision is 2. Scientific means in standard scientific notation, e.g., 5.599e+01.
Convergence	Determines when an assignment to a slot is considered to have changed the value in the slot. When a slot value *V* is given a new value *V'* and | *V* - *V'* | < *V * convergence*, *V* and *V'* are considered to be equal and the assignment is not made. This is necessary to allow the solution of the model to converge. When the value *V* is 0, the convergence criterion is: | *V'* | < *1.1 x 10-12*. Keep in mind that convergence check (as all computation) is done in internal RiverWare units, not in the user (display) units.

Methods

The methods tab is used to select methods associated with a given account. For example, there are categories that have to deal with the calculation of Initial Request for water rights appropriation, Storage Account Carryover, and whether this account has a water right. Methods in the Storage Account Carryover category determine the storage carryover on the "Begin Accrual Date."

Figure 2-12:



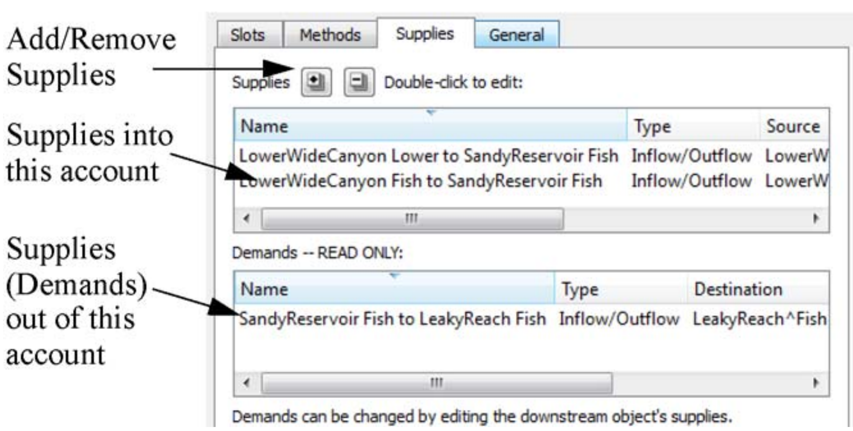
Highlight the desired category row, then use the Selected Method pull down menu to choose an alternative method.

Note: *You can select methods on many accounts in one action using the multi-account method selector which is accessed from the Accounts Manager. See* *"Multiple Account Method Selector" on page 122* *for more information.*

Supplies

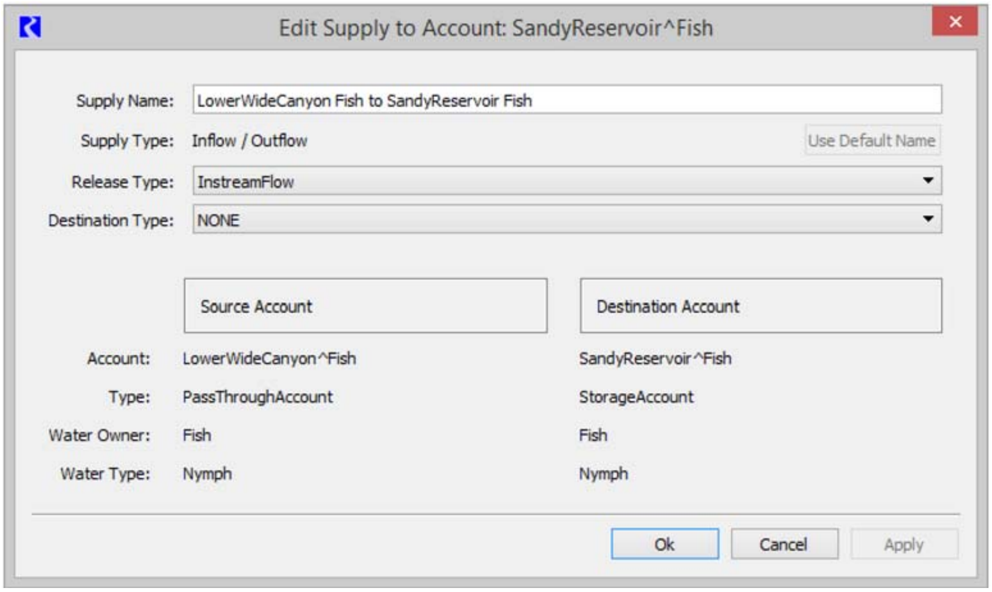
The supplies tab is used to configure supplies into the account. Supplies are the mechanism to link two accounts on the same or linked objects. All supply configurations are done on the downstream account. Supplies are explained further later.

Figure 2-13:



Double-clicking an existing supply will open the `Edit Supply` dialog for that supply. This dialog, shown in the screenshot to the right, can be used to change an individual supply's name, release type or destination.

Figure 2-14:

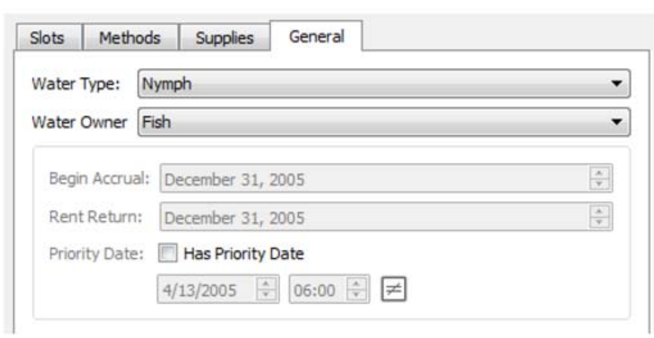


The demands area is used to view the supplies leaving the account. The demands are read-only; nothing can be changed. To edit a demand, the user must open the downstream object and edit the supplies entering the downstream object.

General

The General tab is used to configure general properties of the account. The user can select the Water Type and Water Owner, although the types must already be defined in the Account System Configuration. The user can also specify a `Priority Date` for the account, where applicable. The `Has Priority Date` toggle box must be selected to enter a priority date. The equal/unequal icon indicates if the date shown is unique or not with respect to the priority dates of other accounts. Clicking this icon will open the Account Manager Dialog (see “Water Accounts Manager utility” on page 119) to show accounts with the equal priority date. If equal priority dates are allowed in the Account System Configuration, a copy icon is also shown that when pressed will open an account selector to choose an account from which to copy a priority date. An assigned `Priority Date` is used by rule functions to allocate water.

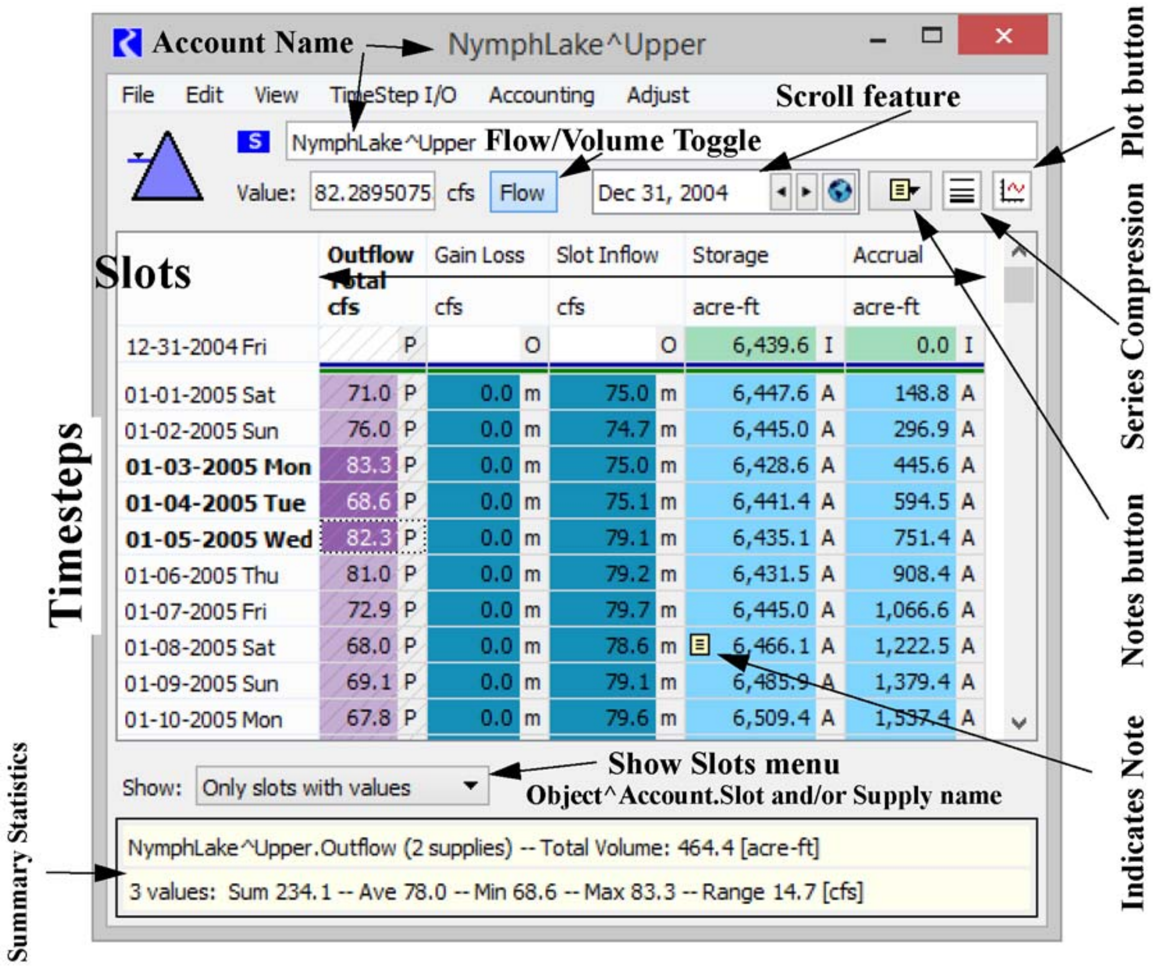
Figure 2-15:



Viewing Account Data through the Edit Account dialog

Let us explore the `Edit Account` dialog. The edit account dialog is used by all account types and look similar although there are different slots for each type of account. The `Edit Account` dialog is the main window used to view an account's data as shown in the following diagram.

Figure 2-16:

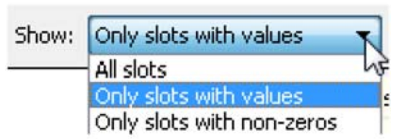


	Outflow Total cfs		Gain Loss cfs		Slot Inflow cfs		Storage acre-ft		Accrual acre-ft	
12-31-2004 Fri		P		O		O	6,439.6	I	0.0	I
01-01-2005 Sat	71.0	P	0.0	m	75.0	m	6,447.6	A	148.8	A
01-02-2005 Sun	76.0	P	0.0	m	74.7	m	6,445.0	A	296.9	A
01-03-2005 Mon	83.3	P	0.0	m	75.0	m	6,428.6	A	445.6	A
01-04-2005 Tue	68.6	P	0.0	m	75.1	m	6,441.4	A	594.5	A
01-05-2005 Wed	82.3	P	0.0	m	79.1	m	6,435.1	A	751.4	A
01-06-2005 Thu	81.0	P	0.0	m	79.2	m	6,431.5	A	908.4	A
01-07-2005 Fri	72.9	P	0.0	m	79.7	m	6,445.0	A	1,066.6	A
01-08-2005 Sat	68.0	P	0.0	m	78.6	m	6,466.1	A	1,222.5	A
01-09-2005 Sun	69.1	P	0.0	m	79.1	m	6,485.9	A	1,379.4	A
01-10-2005 Mon	67.8	P	0.0	m	79.6	m	6,509.4	A	1,537.4	A

On the top of the dialog, the account name is given in the form *Object^Account*. There is an icon for the object type on which this account resides. In this case, it is a Storage Reservoir. In addition, there is an icon that shows the type of this account, i.e. a storage account. On the left side of the dialog are the timesteps for the accounting system. Each column in the table represents a slot that may contain data.

The slots displayed in this dialog depends on the type of account, i.e. storage, passthrough or diversion. Additionally, the `Show` menu allows you to specify whether to show All Slots, Only Slots with values (non NaNs) or Only slots with non-zero values.

Figure 2-17:

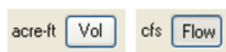


Note: *When first constructing a model, it may be necessary to show all slots to allow the user to input initial storages and accruals. Once constructed, it is frequently desirable to not show empty slots as there are a number of slots on the accounts that likely will not be used in every model.*

- Specify to Show all Slots. Notice that in addition to Outflow, Gain Loss, Slot Inflow, Storage and Accrual, there are now empty columns for Inflow, Diversion, Return Flow, Maximum Accrual, Transfer In, Transfer Out, Begin Year Allocation, and Carry Over.

Above the timesteps, there is a toggle button that switches between flow units and volume units

Figure 2-18:

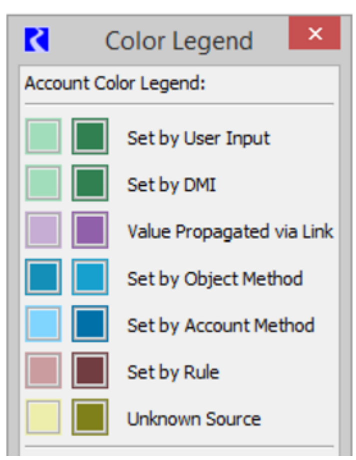


Click on Flow to see the flow values as a volume. Notice that the values have changed and the columns now say acre-feet instead of cfs. The value shown is the volume over the timestep, in this case, acre-feet/day. Slots on accounts use the active Unit Scheme for display and input as described in "Unit Schemes" in Units.pdf. As with the physical system, the accounting system maintains its values in internal units, regardless of the units chosen for display.

Accounting dialogs use colors to represent the process used to set the accounting values. A key is provided that shows the meaning of each color.

- From the menu bar, select View > Show Account Color Legend to bring up a window showing the Grid Cell Legend. The second column of colors is the color when the cell is selected.

Figure 2-19: Screenshot of the accounting color legend



In addition, there is a letter flag on each value that corresponds to each color. Use the key to determine the meaning of each letter. In addition, slots that are Read-Only are shown with a cross-hatch.

Note: *In addition, the "Empty Storage Flag" denoted with the letter "E" and the orange background can be used on outflow supplies from Storage accounts to compute the flow that leads to zero storage. For more information, see "Empty Storage Flag" on page 27.*

In a Rulebased Simulation run, the `Edit Account` dialog also can show the priority of the slot at the given timestep. This can enable/disabled using the `View` > `Show Priorities` menu

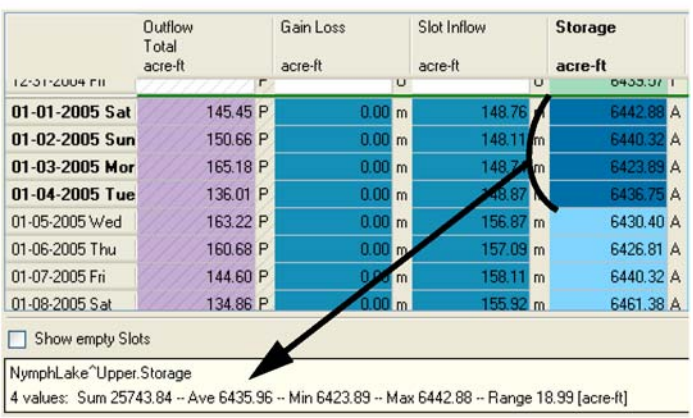
The edit account dialogs show tooltips indicating which rule or DMI was responsible for the specified value, where applicable. Click "Tooltips" for more information.

Back on the `Edit Account` dialog, a scroll button and DateTime spinner allows the user to jump to a specified date. Either type in a valid date in the date box and click the Scroll button to move to that date. Or, use the up and down arrows to advance to the desired timestep.

When the user selects a range of cells in the Edit Account dialog, the dialog displays information about the name and summary information of that selection in the bottom of the window. If the selection is just in one slot, it displays the name of the slot or that the slots consists of one or more supplies. If the selection spans multiple slots, then the name displays how many slots are selected. In addition, summary data at the bottom of the dialog shows the sum, average, minimum, maximum, and range of the selection.

At the bottom of the dialog, the summary information says that the average is 6435.96 acre-ft as shown in the following screenshot. It also displays that the NymphLake^Upper.Storage account is selected.

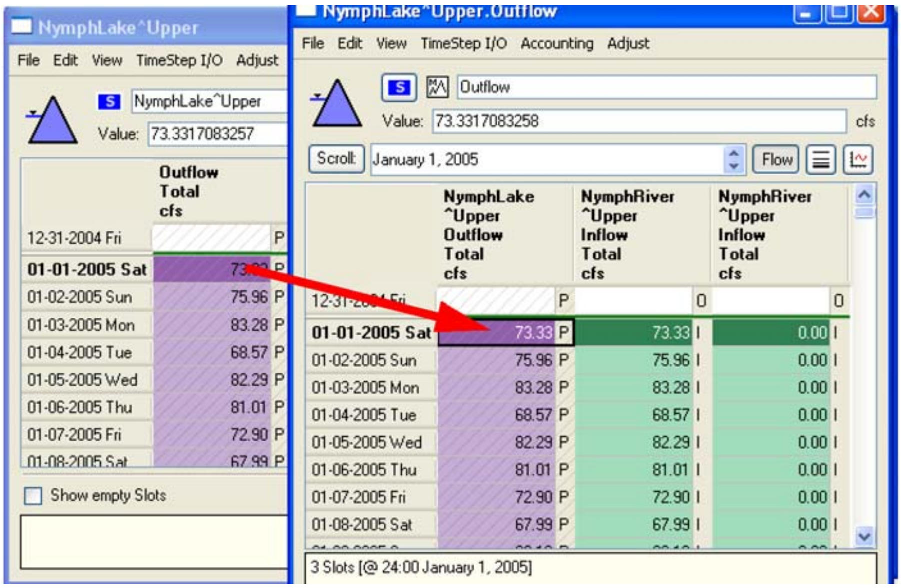
Figure 2-20:



Outflow, Inflow, Transfers, Diversion, and Return Flow slots on the Edit Account dialog can actually consist of more than one value if the user has configured to have multiple supplies coming into or out of that slot. We will discuss this in detail later. For now, let us just be aware that a column can represent more than one time series of data. If a column in the `Edit Account` dialog does in fact have multiple components, the column in the Edit Account dialog will be cross hatched meaning it is read-only. Double clicking on the column opens a new windows showing the multi-slot which consists of the total sum and the component supplies.

A new window opens showing the multislot as shown in the following screenshot. On the left is the Edit Account dialog. On the right is the multi-slot that results when the user double-clicks on the Outflow slot. Note that the numbers in the first column of both dialogs are the same and the second dialog shows the components of the total.

Figure 2-21:



There is also the ability to hide or compress repeated values, a specified value, or NaNs. For more information about Series Display Compression, see "Series Display Compression" in Slots.pdf.

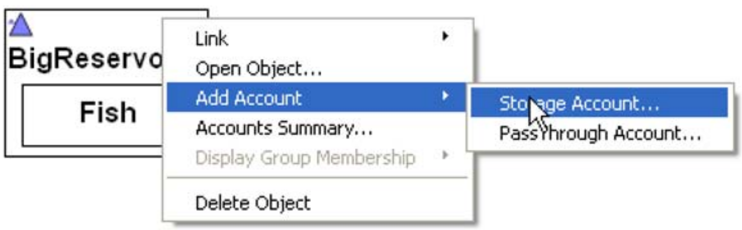
Note: *When using the Edit Account dialog, the user can export a selection of cells using the Export Copy functionality. This allows the user to export data to the clipboard and then paste it into other applications like Excel.*

Creating Accounts

There are several ways to create individual accounts as shown in the following box:

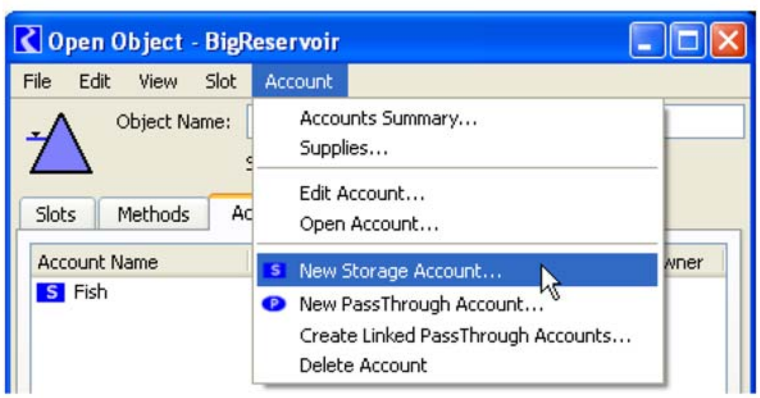
- For non-aggregate objects, right-click on the object in the Accounting View workspace, then choose Add Account and then the desired account type.

Figure 2-22:



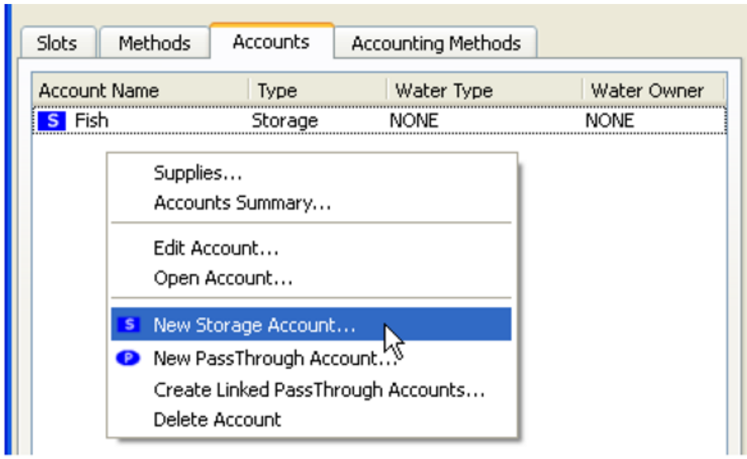
- Within the object dialog, switch to the account tab then select the menu Account > New Storage/PassThrough/Diversion/InstreamFlow Account... If you are on an aggregate object, you will need to highlight one of the elements to which the Account will belong.

Figure 2-23:



- Within the object dialog, switch to the account tab then right-click to see a context sensitive menu from which to choose to create a New Storage/PassThrough/Diversion/InstreamFlow Account... If you are on an aggregate object, you will need to highlight one of the elements to which the Account will belong.

Figure 2-24:



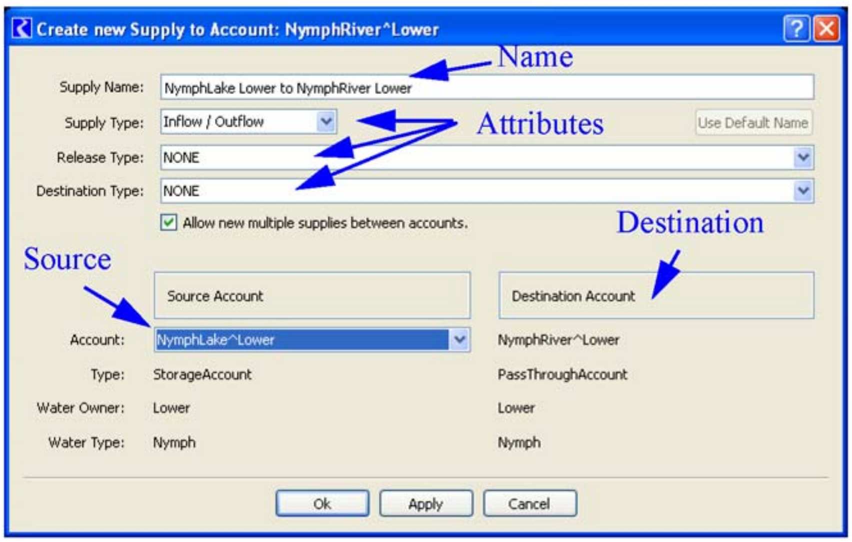
Note: *You can rename accounts en mass from the* Accounts Manager *as described in “Additional Operations in the dialog” on page 120.*

Creating Supplies

There are two ways to create a supply from a "downstream" account to an "upstream" account as shown in the following box:

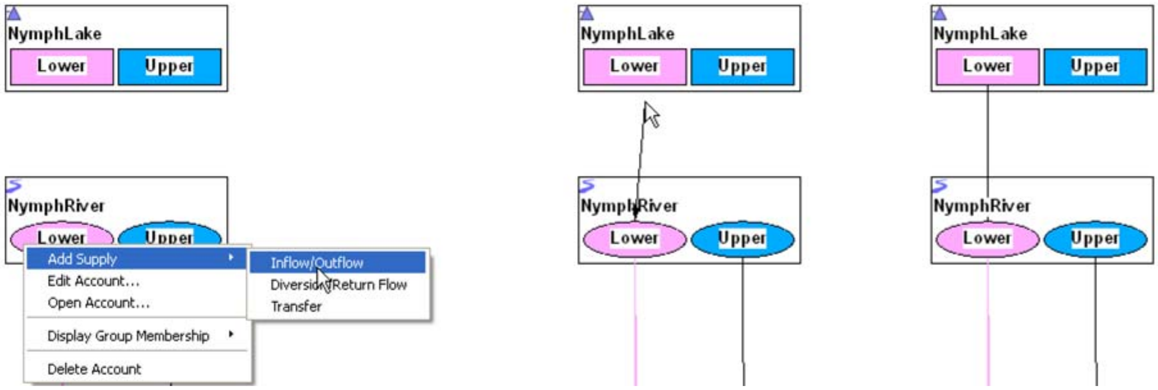
- From the Open Account dialog (Account Configuration), switch to the Supplies tab. Click the  icon to bring up the Create new Supply to Account dialog. Select the source, edit the Supply name if desired, change any attributes and click OK.

Figure 2-25:



- Right-click on the Account in the Accounting View workspace, then choose Add Supply and then the desired supply type. An arrow appears for the user to select the source. Screenshots of these two steps are shown below. Right-click on the source account and create the supply. The default supply name and properties are used. If you desire to change these, you must use either the Edit Supply (from the Supplies tab in the Open Account dialog) or the Supplies Manager to change names or properties. If the source account is more than one link upstream, the Create Linked PassThrough Accounts dialog will appear to create linked Passthrough accounts.

Figure 2-26:



Note: *Before creating supplies, you may wish to modify the default Supply Name Format. See* *"Supply Name Format" on page 129* *for more information.*

Automated Passthrough Account Creation

The Automated Passthrough Account Creation function allows the user to identify two existing Accounts on different Simulation Objects and create linked Passthrough Accounts on all intervening Objects in a single operation. This is a convenient alternative to showing the Object Viewer (or Open Object) dialog box for each of the intervening Objects to create a Passthrough Account. All newly created Accounts and Supplies are assigned properties selected by the user:

- Account Name
- Account Water Type
- Account Water Owner
- Supply Release Type
- Supply Destination (Type)

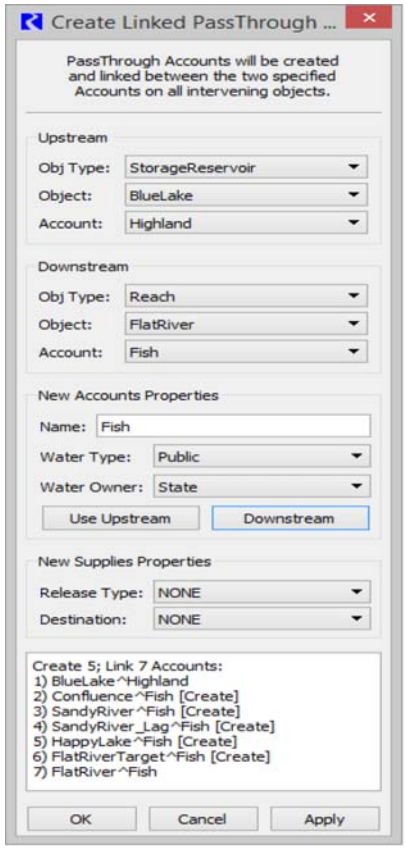
There are several ways to create linked Passthrough accounts:

- Right-click on the downstream account and choose Add Supply > Inflow/Outflow. Move the cursor up to the upstream account and right click again. The Create Linked PassThrough Accounts dialog appears to complete the process. The Upstream and Downstream accounts are automatically filled in.
- Open the object and go to the Accounts tab. Right click in the accounts area (or select Create Linked PassThrough Accounts from the Accounts menu) and choose Create Linked PassThrough Accounts. A dialog appears to complete the process. On an aggregate object, you must first select an element.
- Open the object and go to the Accounts tab. Select Create Linked PassThrough Accounts from the Accounts menu. A dialog appears to complete the process.
- From the Accounts Manager, select Account > Create > Passthrough Accounts to open the Create Linked PassThrough Accounts dialog.

Quick Usage Overview

1. Open the Create PassThrough Accounts dialog box from one of the methods specified above or:

Figure 2-27:



2. Select the correct Upstream Account
    - Select an Object Type (or “All Object Types”)
    - Select an Object of that Type
    - Select an Account on that Object
3. Select the desired Downstream Account
    - Select an Object Type (or “All Object Types”)
    - Select an Object of that Type
    - Select an Account on that Object
4. Enter the desired Account Name for newly created Accounts
5. Enter the desired Water Type for newly created Accounts
6. Select the desired Water Owner for newly created Accounts
7. Select the desired Release Type for newly created Supplies
8. Select the desired Destination (Type) for newly created Supplies
9. Press either the ‘OK’ or ‘Apply’ button.

Detailed Usage Information

The following sections show detailed information on using the dialog:

- "Invoking the Dialog."
- "Overview of the Dialog."
- "Account Selector Boxes" on page 116.
- "New PassThrough Account property specification" on page 116.
- "New Supply Property specification" on page 117.
- "Enabledness of the OK and Apply Buttons" on page 117.
- "Selection Status Area" on page 117.
- "Reporting Account and Supply Creation Results" on page 118.

Invoking the Dialog

This dialog box can be shown using either of these two menu operations:

- From the Water Accounts Manager dialog box:

  Account -> Create -> PassThrough Accounts...

- From an object dialog:

  Account -> Create -> PassThrough Accounts...

Overview of the Dialog

The Create PassThrough Accounts dialog box includes the following user controls:

1. Two Account Selector Boxes:
   - An Upstream Account (Selections: Object Type, Object, Account)
   - A Downstream Account (Selections: Object Type, Object, Account)
2. New Account Properties controls for specifying properties given to the Accounts created with by operation.
   - An Account Name (Text Entry Field)
   - A Water Type (Option Menu selection)
   - A Water Owner (Option Menu selection)
3. Optional push button for assigning the Account Properties (Name, Water Type and Water Owner) of the currently selected Upstream or Downstream Account to the New Account property controls described in the previous item.
4. New Supplies Properties controls for specifying properties given to the Supplies created by this operation.
5. A multi-line text field for display of either:

- the reason the entered and selected values do not specify a valid creation operation, or
- a list of the Accounts which will be created and linked (via new Supplies) if the user presses the 'OK' or 'Apply' buttons.

6. Push buttons: 'OK', 'Apply', and 'Close'

Account Selector Boxes

The Create PassThrough Account dialog box has two Account Selection boxes used to indicate an Upstream Account and a Downstream Account. Each has three option menus:

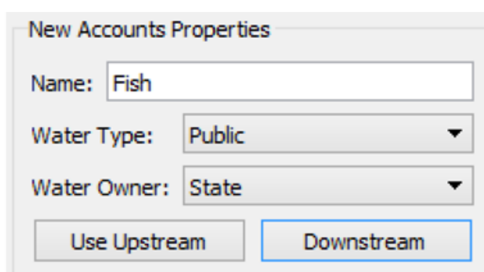
Figure 2-28:

- `Obj (Object) Type` option menu**:** The user selects either "All Object Types" or one of the Object Types for Objects which can have Accounts and which are represented in the model.
- `Object` option menu: The user selects one of the Simulation Objects in the model of Object Type indicated in the Obj Type option menu.
- `Account` option menu: The user selects one of the Accounts in the Simulation Object indicated in the Object option menu.

New PassThrough Account property specification

All the PassThrough Accounts created by a Create PassThrough Accounts operation are given the same **Name, Water Type** and **Water Owner.** (It's OK for all of the Accounts to have the same name because only one such Account will be created on any particular Object).

Figure 2-29:



These properties can be those of the selected Upstream Account, or those of the selected Downstream Account, or any arbitrary valid name and property selection.

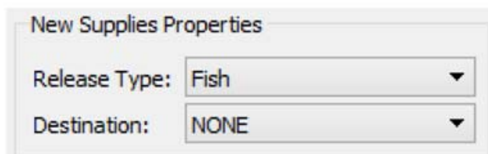
The name appearing in the text entry field -- which can be freely edited by the user -- is the name to be used for all new created Accounts. Pressing the “Use Upstream” or “Downstream” buttons assigns the Name, Water Type, and Water Owner of the currently selected Upstream or Downstream Account to those three entry / selection widgets. The user can then make modifications to any of those values.

Valid values for Water Type and Water Owner are defined by the user in the Accounting System Configuration.

New Supply Property specification

Newly creates Supplies (links between Accounts) will be given the default name (based on the names of the two linked Accounts) and the following properties.

Figure 2-30:



- Release Type
- Destination (Type)

Valid values for these two properties indicated above are defined by the user in the Accounting System Configuration.

Enabledness of the OK and Apply Buttons

The “OK” and “Apply” push buttons are enabled only if all of the following conditions are met.

- The selected **Upstream Account can accept a Demand** to a downstream PassThrough Account (or to the Downstream Account, if that is on an adjacent, physically linked Object).
- The selected **Downstream Account can accept a Supply** from an upstream PassThrough Account (or from the Upstream Account, if that is on an adjacent, physically linked Object).
- **A path of physical links exists** between the Upstream Object and the Downstream Object which are needed for the creation of Supplies connecting the indicated Accounts via new PassThrough Accounts. To determine this state, the mechanism searches for a “downstream” linked path from the Upstream Object to the Downstream Object.
- The **intervening Objects do not yet contain** an Account with the name to be used for the new PassThrough Accounts.
- The intervening Objects are of Object Types on which **PassThrough Accounts can be created.** Note that this restriction does not apply to the Upstream or Downstream Objects. That is, the selected Accounts don’t have to be PassThrough Accounts.

Selection Status Area

In addition to the enabledness of the “OK” and “Apply” push buttons, a brief message about the validity of the

Create PassThrough Account operation, given the current input selections, is displayed in the Selection Status Area.

Status Area text messages include:

- “No Upstream Account specified.”
- “No Downstream Account specified.”
- “Upstream and Downstream Accounts must be distinct.”
- “Upstream and Downstream Objects must be distinct.”
- “Account ‘<account name>’ already exists”
- “No physical path exists from Upstream Object to Downstream Object.”

If the selections specify a valid Create PassThrough Accounts operation, then the Accounts which would be created and linked (with Supplies) are indicated. (See picture, above).

Reporting Account and Supply Creation Results

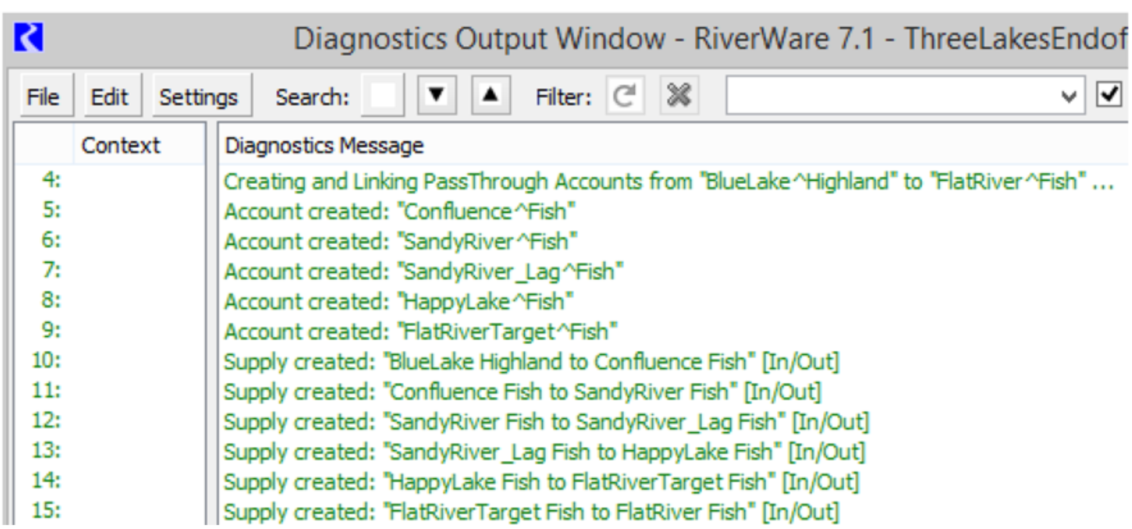
During the course of the Create PassThrough Accounts operation (as a result of pressing the ‘OK’ or ‘Apply’ buttons), any detected problems will be reported as an Error in the Diagnostics Output window.

Note that this does not apply to the error conditions detected BEFORE the ‘OK’ or ‘Apply’ button is pushed -- those conditions actually prevent those button from being pushed.

The following actions are reported as Informational messages in the Diagnostic Output window:

- Creating and Linking PassThrough Accounts from “<upstream account>” to “<downstream account>”
- Account created: “<account name>”
- Supply created: “<supply name>” [<supply type>]

Figure 2-31:



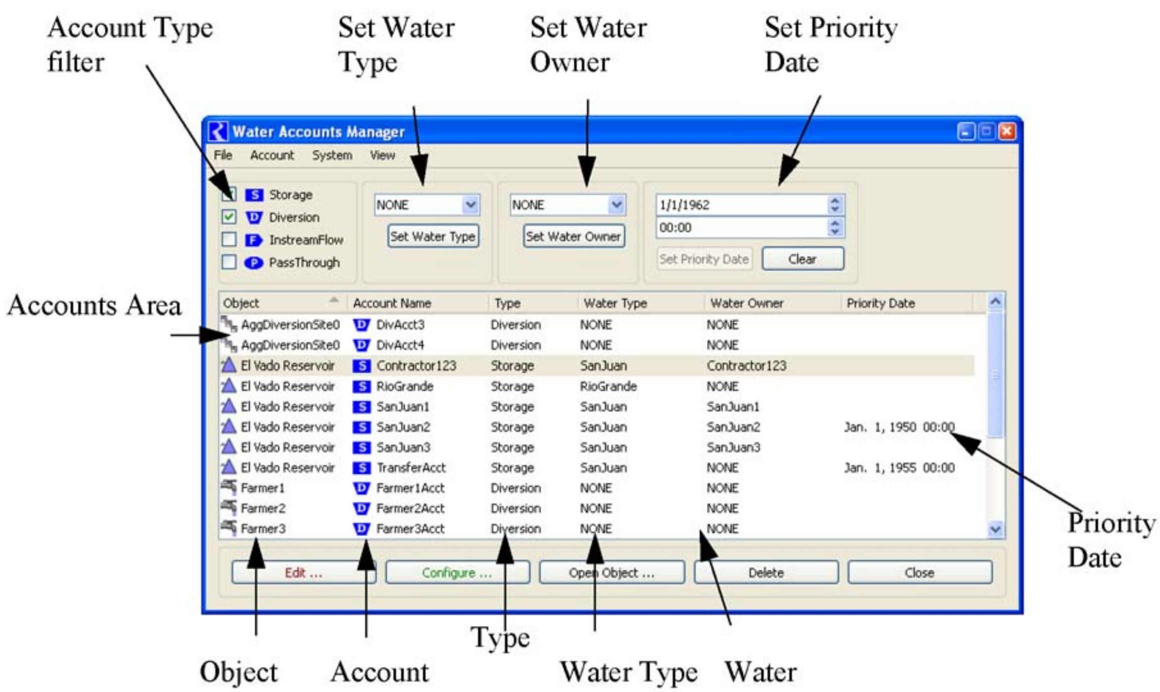
Water Accounts Manager utility

The Water Accounts Manager dialog displays some or all of the accounts in the model. It is an alternative way to access accounts in the model for purposes of viewing and configuring.

Tour of the Water Accounts Manager

The `Water Accounts Manager` is accessed from the `Accounting` > `Accounts Manager` menu on the main RiverWare workspace. There is only one Water Accounts Manager for a given model. Following is a screenshot that shows the dialog with key areas highlighted:

Figure 2-32:



The `Water Accounts Manager` displays one account per row. Sortable columns display the Object, Account Name, Water Type, Account Type, Water Owner, and Priority Date, if applicable. Following is a description of the various buttons, toggles, and features of the utility:

Account Type Filter

Each Account type (Storage, Diversion, Passthrough, and InstreamFlow) has a toggle button. If the toggle is selected, then accounts of that type are shown in the Accounts Area.

Set Water Type

This button is used to set the Water Type of many accounts at one time. Highlight the desired accounts, select a new Water Type from the pull down menu, and click `Set Water Type` to make the change.

Set Water Owner

This button is used to set the Water Owner of many accounts at one time. Highlight the desired accounts, select a new Water Owner from the pull down menu, and click Set Water Owner to make the change.

Set Priority Date

This button is used to set the priority date of an account. If each account must have a unique priority date, the user can only change one priority date at a time. To do this, select the desired account, change the priority date by either typing a new date in the box or clicking the up or down arrow, and click Set Priority Date to apply. If the accounting system configuration allows equal priority dates among accounts ("Account System Configuration" on page 92), multiple accounts can be selected and changed at the same time. Accounts with the same priority date will be distinguished by a background color shading of the priority date in the accounts list.There is also a Clear button to remove priority dates from one or more accounts.

Select Account Methods: Access the Multiple Account Method Selector to select methods on many accounts in one action using the Account > Account Method Selector... This is described in "Multiple Account Method Selector" on page 122.

Additional Operations in the dialog

The Water Accounts Manager can also be used to access other dialogs in the accounting system.

Edit Account

Select the desired account(s) and click the Edit Account button to edit one or more accounts. Also, double click on an account row and the Edit Account dialog opens.

Configure Account

Select the desired account(s) and click Configure Account to configure one or more accounts.

Rename Accounts:

Select the desired account(s) and click Account > Rename to change the name of many accounts at once. If there are any accounts with that name already on the object, a message will be posted and you will not be able to rename.

Open Object

Select the desired account(s) and click Open Object to open the object containing the account.

Delete

Delete the selected account.

Sort

Accounts in the dialog can be sorted in two methods: Click on the column heading to sort according to that attribute. Click it again to reverse the order. Also, the View menu can be used to sort accounts in various ways.

Open additional dialogs

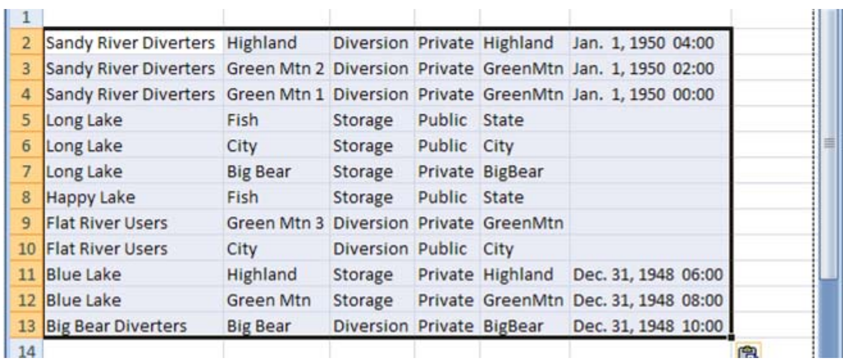
The Account and System menus can be used to access other dialogs in the accounting system including the Object Account Summary, Account System Configuration, Exchange Manager, Supply Manager, and Object Level User Defined Accounting Method set editor.

Copy List to Clipboard

The menu operation `File` > `Copy List to Clipboard` copies a "tabular" string representation of the currently displayed Account list to the System Clipboard. Fields are delimited with Tab characters, lines are delimited with New Line characters -- suitable for importing into a spreadsheet, such as Microsoft Excel.

Figure 2-33:

Figure 2-34:

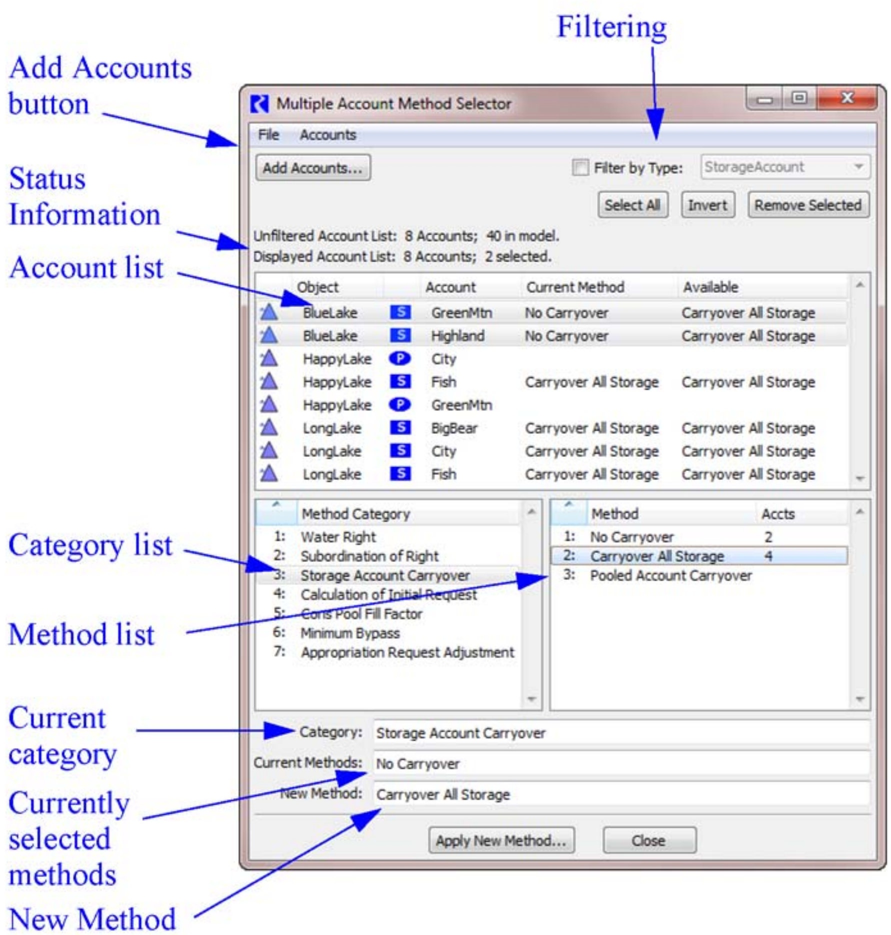


1						
2	Sandy River Diverters	Highland	Diversion	Private	Highland	Jan. 1, 1950 04:00
3	Sandy River Diverters	Green Mtn 2	Diversion	Private	GreenMtn	Jan. 1, 1950 02:00
4	Sandy River Diverters	Green Mtn 1	Diversion	Private	GreenMtn	Jan. 1, 1950 00:00
5	Long Lake	Fish	Storage	Public	State	
6	Long Lake	City	Storage	Public	City	
7	Long Lake	Big Bear	Storage	Private	BigBear	
8	Happy Lake	Fish	Storage	Public	State	
9	Flat River Users	Green Mtn 3	Diversion	Private	GreenMtn	
10	Flat River Users	City	Diversion	Public	City	
11	Blue Lake	Highland	Storage	Private	Highland	Dec. 31, 1948 06:00
12	Blue Lake	Green Mtn	Storage	Private	GreenMtn	Dec. 31, 1948 08:00
13	Big Bear Diverters	Big Bear	Diversion	Private	BigBear	Dec. 31, 1948 10:00
14						

Multiple Account Method Selector

The Multiple Account Method Selector allows you to select account level methods for many accounts in one action. The dialog is similar to the Multi-Object Method Selector that is described in "Multiple Object Method Selector" in ObjectDialogs.pdf.

Figure 2-35:



Quick Usage Overview

- Open the Water Accounts Manager and optionally select the desired accounts.
- From the Water Accounts Manager use the Account > Account Method Selector... menu
- The dialog initially shows the accounts that were selected in the Accounts Manager. Add accounts by clicking the Add Accounts... button, then select accounts in the selector and click Ok.
- Optionally, filter the accounts by type.
- Highlight the desired accounts to which a new method should apply.

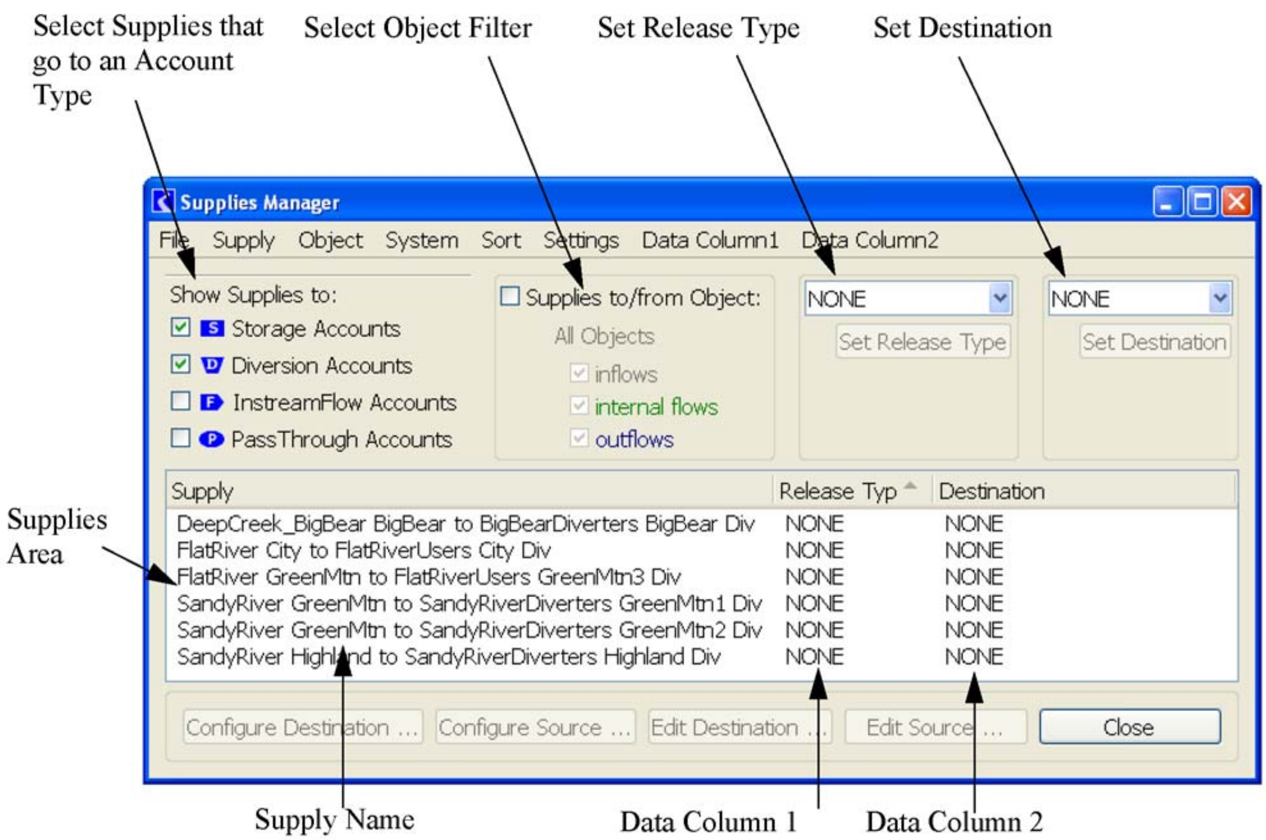
- Select a category from the Method Category list.
- Select a new method from the Method list.
- Verify that the selected method is available on all desired accounts by inspecting the Available column in the accounts list.
- Apply the new method to the highlighted accounts by clicking the Apply New Method button.
- Verify that the accounts listed in the confirmation dialog are the accounts you intend to change, and press OK.
- The accounts will now have the selected methods.

Supplies Manager

The Supplies Manager dialog box displays supplies within the model. The main part of the Supplies Manager dialog box presents one Supply per row.

The Supplies Manager is accessed from the Accounting > Supplies Manager menu on the main RiverWare workspace. There is only one Supplies Manager for a given model. Following is a screenshot that shows the dialog with key areas highlighted:

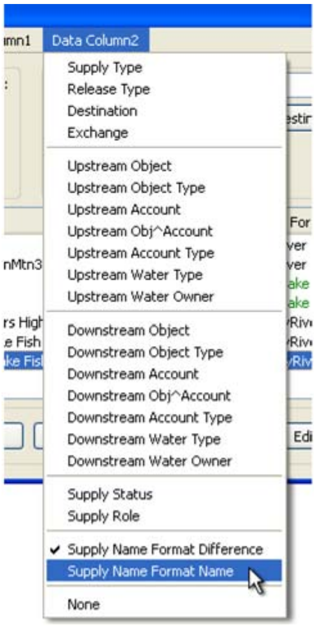
Figure 2-36:



Supplies Manager “Columns”

The Supplies Manager List presents one Supply per row, with three columns of information about the Supplies. The width of the columns can be adjusted by “dragging” the line separating the column headings.

Figure 2-37:



The first of the three columns is always the Supply Name (which can be quite long). The content of the other two columns is Configurable by the user through the Data Column1 and Data Column2 menu bar items. Within these two menus is the option to select the data display as shown in the screenshot on the right. The selected variable has a check mark next to it.

Color Annotation of Supply Rows

Supplies that represent **internal transfers** between two accounts on the same object are shown in **Green** text. Supplies between Accounts on different Objects which are **not directly physically linked** (technically, this is an error condition) are shown in **Red** text. Currently, if the upstream and downstream Objects are linked together on *any slots,* the Supply is shown to be normal (not Red), even if the slots used for the link(s) don't make sense. (It is the user's responsibility to insure that the links make sense).

When a particular Object is selected for filtering (see next section), Supplies that represent Outflows from that object are shown in **Blue** text. (Supplies which represent Inflows to the Object are shown in **Black**).

Supply Filtering

The Supplies Manager can show all the Supplies in a model or a filtered subset of Supplies. The user can filter supplies by the type of the destination account. This is done using the Show Supplies to: toggle area and selecting one or more of the following types: Storage, passthrough, instream flow, and/or diversion accounts.

Also, the Supply Manager can present only the Supplies related to a particular simulation object. If filtering by an object is specified by checking the Supplies to/from Object, the list of Supplies shown is limited to those which represent inflows (into the object) and/or outflows (from the object) and/or internal flow (between two accounts on the object). The internal flow toggle and the outflows toggle are shown respectively in Green and Blue to correspond to the color annotation of supply rows.

The first time the user toggles on the Supplies to/from Object, an object selector dialog appears. The user can also select Object > Select from the menu bar to select a different object using the selector. Note, only one object can be selected. When the Supplies to/from Object is off, the object label reads "All Objects" and the inflows/ internal flows/outflows toggles are turned on, but the controls are greyed out.

Supply Sorting

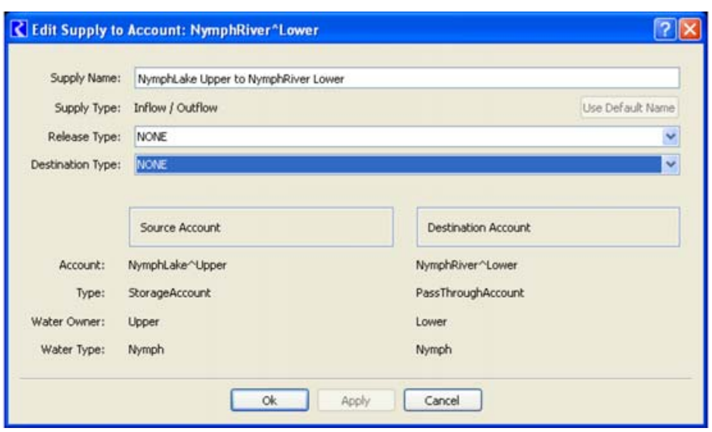
The supply rows can be sorted based on any of the column headings. The user can sort the rows by any of the three displayed columns by clicking on the column header. Also, the user can sort the supply rows by selecting any of the criteria under the Sort menu item. This can be useful because the user can sort by a variable but it is not necessary to show that variable as one of the data columns. In general, these values are self explanatory, except for Supply Status. The sort by Supply Status operation sorts the Supply Rows in three or four groups, and by supply name within each of those groups:

- Supplies between Accounts on distinct Objects which do not have a direct physical link are shown in **Red**. Click "Color Annotation of Supply Rows" for more information).
- Supplies between Accounts on the same object are shown in **Green**.
- Other supplies between accounts on distinct objects (having a direct physical link) are shown in **Black** or **Blue**. Only if a particular object is selected for filtering, the remaining supplies are separated into two groups:
  - Supplies which represent inflows into the object (shown in **Black**).
  - Supplies which represent outflows from the object (shown in **Blue**).

Operations on Selected Supplies

Double-clicking a row in the Supply Manager will open the Edit Supply dialog for that Supply. This dialog, shown in the screenshot to the right, can be used to change an individual supply's name, release type or destination.

Figure 2-38:



Clicking on a row in the supplies area causes it to be selected. Multiple-row selection are also supported for operations that can apply to more than one supply, e.g. deletion or setting the Release Type or Destination.

Set Release Type

This button is used to set the Release Type of one or more supplies. Highlight the desired supply(ies), select a Release Type from the pull down menu, and click on Set Release Type to make the change.

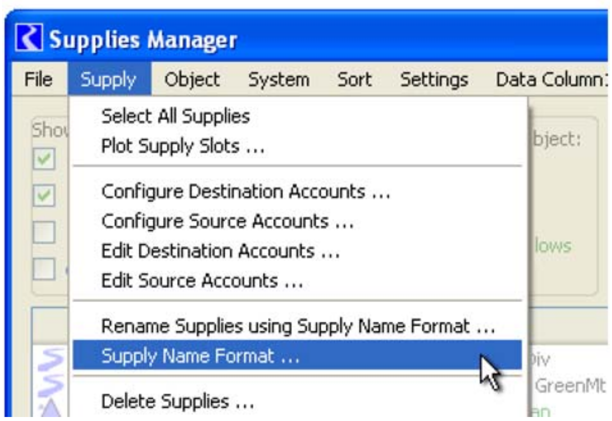
Set Destination

This button is used to set the Destination of one or more supplies. Highlight the desired supplies, select a Destination from the pull down menu, and click Set Destination to make the change.

Several operations are also available as buttons in a row along the bottom of the dialog box. Currently these are the Configure Destination, Configure Source, Edit Destination, Edit Source and Close (dialog box) operations. The operations on Supplies are accessed from the Supply menu. They are defined as follows:

Select All Supplies -- Select all supplies listed in the dialog.

Figure 2-39:



Plot Supply Slot... -- Creates a plot of the selected supplies.

Configure Destination Account... -- Open up the Open Account dialog for the supply's destination (downstream) account, for each of the selected supplies.

Configure Source Account... -- Opens the Open Account dialog for the supply's source (upstream) account for each of the selected supplies.

Edit Destination Account... -- Opens the account editor for the supply's destination (downstream) account for each of the selected supplies.

Edit Source Account... -- Opens the account editor for the supply's source (upstream) account for each of the selected supplies.

Supply Name Format... -- Configured the format to use for supply names. See "Supply Name Format" on page 129 for more information.

Rename Supplies using Supply Name Format... -- Rename the selected existing supplies using the currently defined Supply Name Format. Use the menu Edit Supply Name Format to configure the format. Then choose this menu to apply that format. A confirmation dialog box is shown when the user selects the "Rename Supplies using Supply Name Format ..." operation.

Delete Supply... -- Deletes the selected supplies; the user is given a dialog box to confirm the action.

Operations on Objects

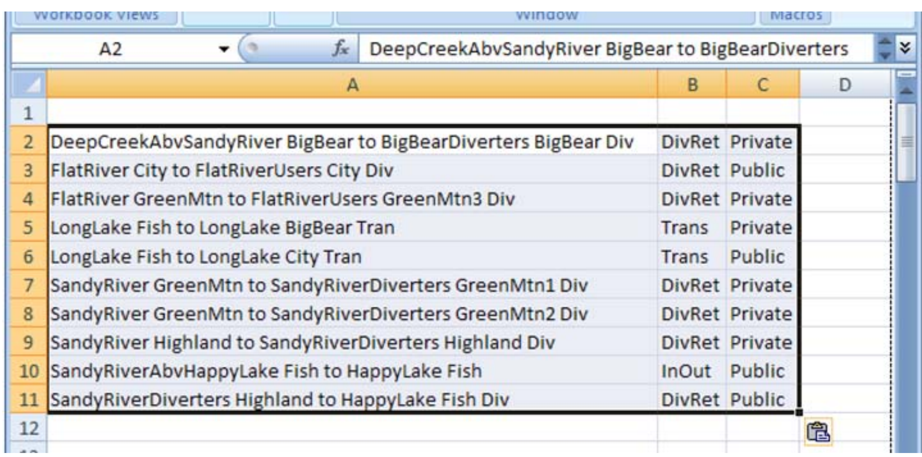
From the Object menu, operations are available on the destination object of the selected supply. The user can bring up either the Object Viewer (or Open Object) dialog box (in the Accounts tab) or the Object Account Summary dialog box (showing the sums of series slots across a set of Accounts on a single object) for the selected object. In addition, accounting system dialogs can be accessed using the System menu including: Account System Configuration, Exchange Manager, Accounts Manager, and Object Level User Defined Accounting Method set editor.

Copy List to Clipboard

The menu operation File > Copy List to Clipboard copies a "tabular" string representation of the currently displayed Supply list to the System Clipboard. Fields are delimited with Tab characters, Lines are delimited with New Line characters -- suitable for importing into a spreadsheet, such as Microsoft Excel.

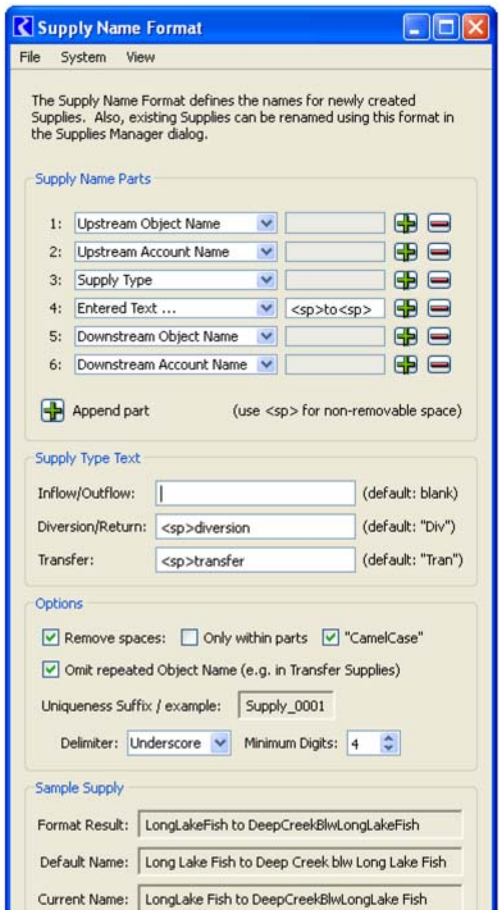
Figure 2-40:

Figure 2-41:



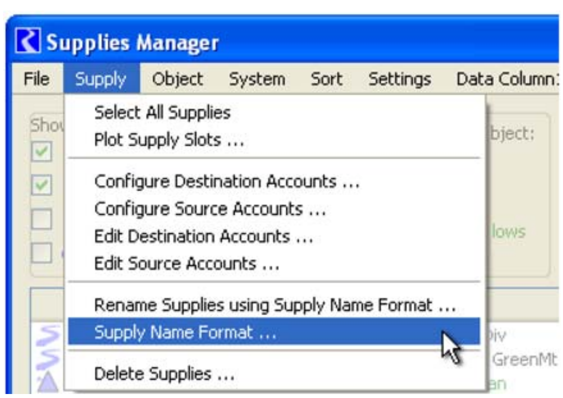
Supply Name Format

Figure 2-42:



A RiverWare model maintains a single instance of a “Supply Name Format” specification. This is editable in the Supply Name Format dialog -- accessible from the Supply Manager’s `Supply` menu:

Figure 2-43:



Changes made in the Supply Name Format are applied immediately, but can be canceled (reverted to the settings which were active when the dialog was shown) by clicking the `Cancel Changes` button.

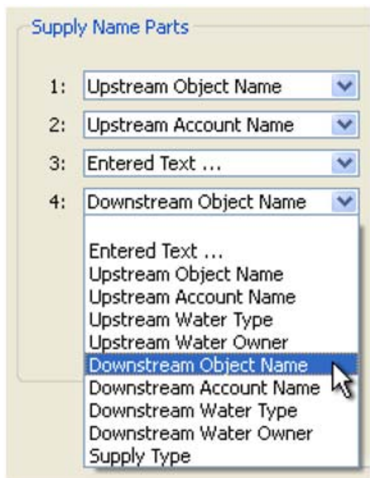
The user can also restore RiverWare Default settings by clicking the `Restore Default` button. The default format is described in "Default Supply Name Format" on page 135.

Those two buttons are independently disabled if the edited settings already match these operations' respective settings. The user can effectively toggle between the default settings and the current settings (those which were active when the dialog is shown) by clicking back and forth between the "Restore Defaults" and "Cancel Changes" buttons -- assuming that those settings are different.

The `Close` button dismisses the Supply Name Format dialog.

Supply Name Part sequence

Figure 2-44:



In the new Supply Name Format dialog, the user defines a sequence of "Supply Name Parts" made up of the following entities:

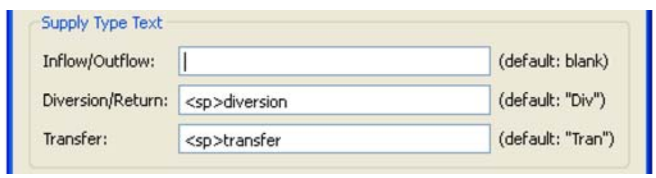
- Upstream Object Name
- Upstream Account Name
- Upstream Account Water Type
- Upstream Account Water Owner
- Downstream Object Name
- Downstream Account Name
- Downstream Account Water Type
- Downstream Account Water Owner
- Supply Type Text (configurable in frame below)
- Any entered text having valid supply name characters (e.g. "to")

The user can insert, append and remove Supply Name Parts (rows) using the Plus and Minus icon buttons. Up to 12 parts can be defined:

Figure 2-45:

The user can provide Supply Type Text strings for each of the three types of supplies:

Figure 2-46:



- Inflow/Outflow
- Diversion/Return (default is “Div”)
- Transfer (default is “Tran”)

Allowed characters in the “Entered Text” and “Supply Type Text” fields are letters (upper and lower case), digits, spaces and underscore characters. Also, a “non-removable space” tag (“<sp>”) can be entered to specify spaces which are not to be removed by the space removal and “CamelCasing” features (with some exceptions). (Other ‘<‘ and ‘>’ characters are silently removed from the final computed name).

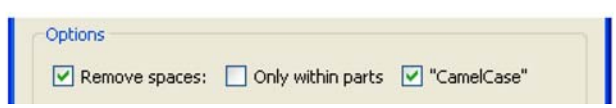
Supply Name Generation Options

Option: Part Assembly, Space Removal and Camel Casing

The sequence of Supply Name Parts is assembled into a single string, with the parts initially separated by spaces. Multiple contiguous spaces are condensed into single spaces, and spaces are removed from the beginning and end.

The user can optionally specify that spaces be removed, either only *within* the individual parts or also in between the separate parts.

Figure 2-47:



If “CamelCasing” is enabled, wherever a space is removed, if the subsequent character is a lower-case letter, that letter is changed to upper-case.

The translation of non-removable space tags (“<sp>”) occurs *after* the application of optional space removal and camel casing. However, all *contiguous* spaces (including “<sp>” tags) are condensed into a single space, and spaces are removed from the beginning and end of the generated name.

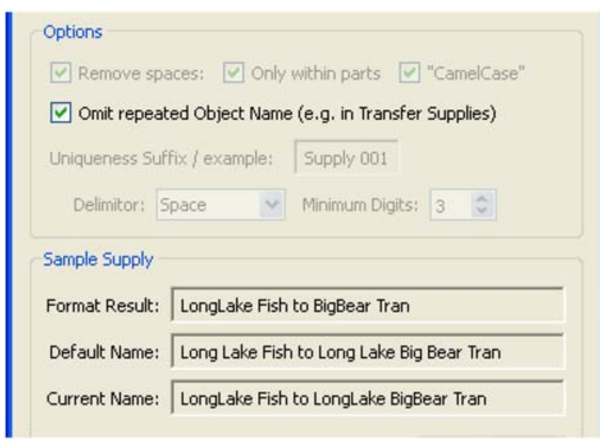
In the five examples below, the non-removable spaces around the “to” text part are preserved. Also, the space tag preceding “to” prevents that word from being “camel cased”. Otherwise, “to” would have been changed to “To” in the cases where camel casing is enabled. In the examples below, the effect of camel casing is illustrated with the treatment of “abv” (an abbreviation for “above”).

Figure 2-48:

Option: Omit Repeated Object Name (e.g. in Transfer Supplies)

In RiverWare, transfer supplies represent the movement of water from one account on a simulation object to another account on the **same** object. The “Upstream Object” and the “Downstream Object” are the same. The default supply naming convention results in that object being included twice.

Figure 2-49:

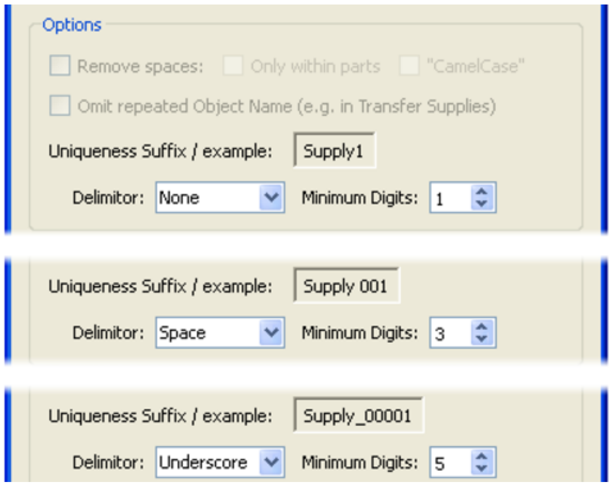


The user has the option of omitting repeated object names in generated supply names by clicking on the `Omit repeated Object Name` toggle.

Option: Supply Name Uniqueness Suffix

All supply names in a RiverWare model must be unique and different than other simulation objects. It's possible that a Supply Name Format will not generate unique names of supplies. Only in such cases, a numeric suffix is appended to the generated supply name.

Figure 2-50:



The numeric "uniqueness" suffix generated for supplies, as needed, can be configured by the user in the following ways.

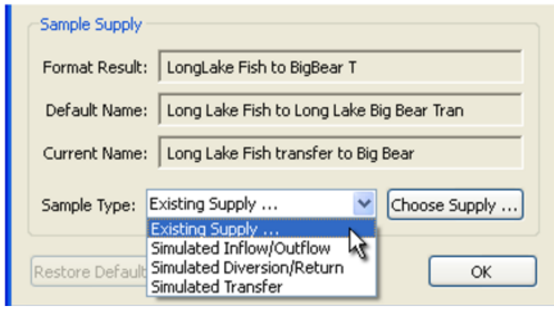
- The suffix can optionally start with a space or underscore Delimiter.
- The number field can have a fixed number of digits, padded with leading zeros. This is provided as a *minimum* number of digits -- more digits will be used for any particular supply if the indicated digit count isn't sufficient.

The dialog dynamically illustrates the effect of these two settings in the "example" field (see above). The effect is also illustrated within the Sample Supply box Format Result field, when needed for the application of the format to the chosen sample supply (see below). Generated supply names are made unique not only with respect to all existing supplies, but also with respect to all other top-level objects, including simulation objects (reservoirs, reaches, etc) and exchanges.

Supply Sample Box

This Supply Name Format dialog shows the result of the application of the current format to a single "sample" supply. The sample supply can be an actual existing supply OR one of three "simulated" (fake) supplies -- one for each of the three Supply Types. The Sample Supply box shows three versions of the Supply name:

Figure 2-51:

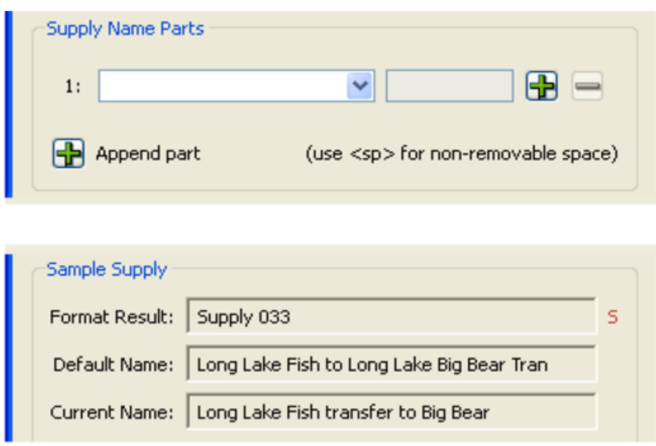


- **Format Result:** the name of the sample supply as computed using the current format.
- **Default Name:** the name of the sample supply, as computed using the default.
- **Current Name:** the current name of the sample supply (shown only when an actual existing supply is used as the sample).

Note: *This dialog does not currently support renaming the sample supply using the configured format. Supply renaming can be done in the Supply Manager, which is described in “Operations on Selected Supplies” on page 126.*

The Format Result field will show a “uniqueness suffix” if that is required for the particular sample supply. (In the case of the “simulated” samples, a uniqueness suffix is shown if the Supply Name Part list is empty or all blank).

Figure 2-52:

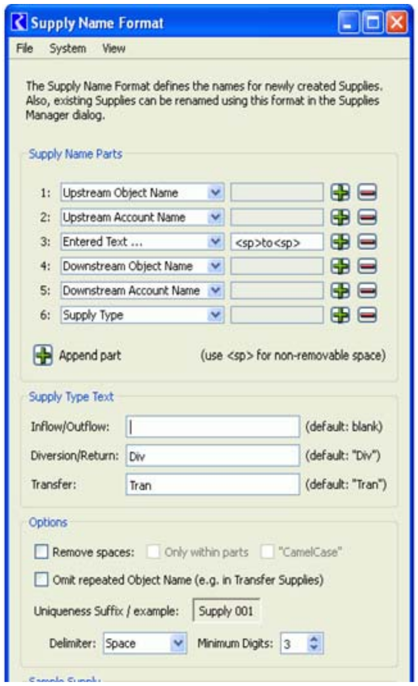


When a uniqueness suffix is applied to the generated name, a RED “S” character is shown to the right of the field.

Figure 2-53:

Default Supply Name Format

Figure 2-54:



The default Supply Name Format is illustrated in the image to the right. It contains these Supply Name Parts and does not perform space removal:

<Upstream Object> <Upstream Account> “ to ” <Downstream Object> <Downstream Account> <Type>

where <Type> is one of the following:

- for Inflow/Outflow Supplies: (blank)
- for Diversion/Return Supplies: “Div”
- for Transfer Supplies: “Tran”

Supply Name Format Clipboard

A single instance of a Supply Name Format can be saved in a special “clipboard”. The clipboard format persists between RiverWare sessions, and can be used to copy a Supply Name Format from one RiverWare model to another.

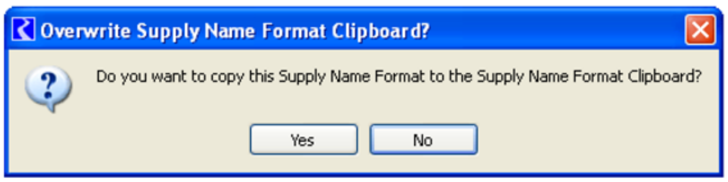
Figure 2-55:

The clipboard functions are under the File menu:

- Paste from Supply Name Format Clipboard
- Copy to Supply Name Format Clipboard

The Copy operation is confirmed with the following dialog:

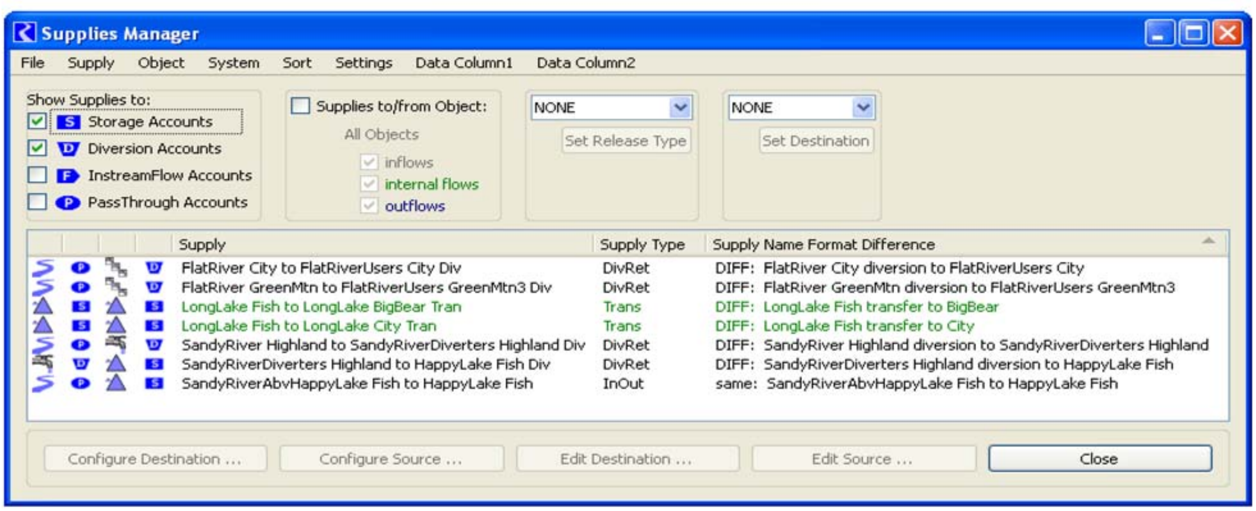
Figure 2-56:



Showing Supply Format Differences

The Supply Manager has the ability to show the names computed for each of the existing supplies using the model's current Supply Name Format -- optionally with an indication that the computed name is different than the supply's current name -- see example below. To do this select the `Supply Name Format Difference` in either `Data Column 1` or `Data Column 2` menus. The string "DIFF" indicates the supply is different than the Supply Name Format. The string "same" indicates they are identical.

Figure 2-57:



Object Account Summary

The Object Account Summary displays summary information for the accounts on a given object. This information is useful to see summary information about all or some of the selected accounts on a given object (or possibly objects in the future). For example, the user could select to see the total accounting storage for a given water type. Or, the user could see the total volume of water released downstream from all accounts on an object and compare it to the Object's simulation Outflow to ensure that the two systems are reconciled.

Accessing the Object Account Summary dialog

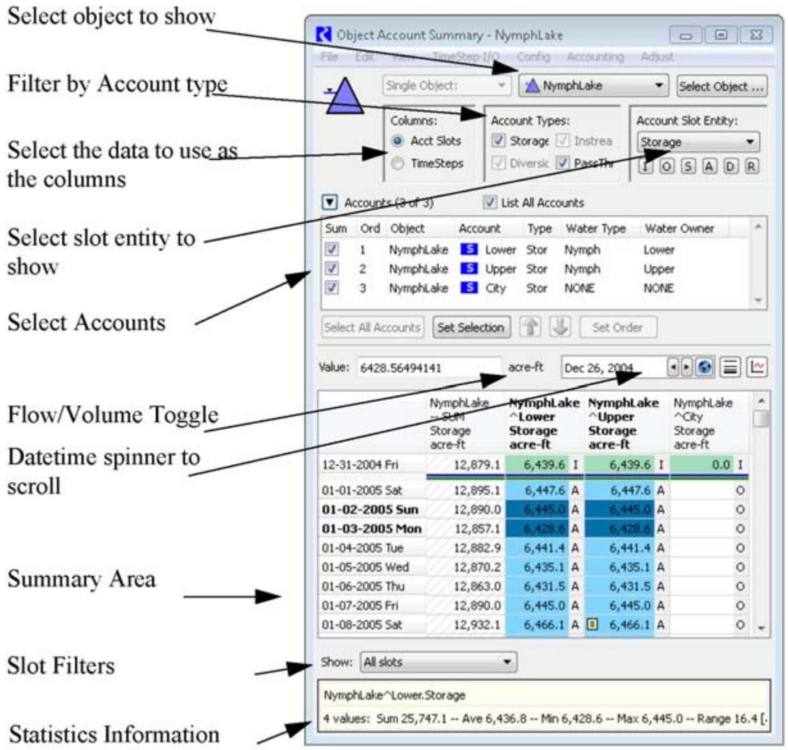
To open the Object Account Summary dialog, use one of the following options:

- From the Object Viewer (or Open Object) dialog, use the Account > Accounts Summary... menu option.
- From the Accounting View of the workspace, right-click on a simulation object and select Accounts Summary... from the context menu.
- From Water Accounts Manager, select Accounts > Object Summary.
- From the Supplies Manager, select Object > Summary

Tour of the Dialog

Following is a screenshot showing the Object Account Summary dialog with the major components highlighted.

Figure 2-58:

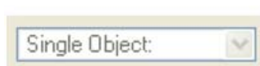


The dialog is used to filter the information to show. In general, the top part of the dialog contains toggles and options which the user can select to control the information shown. The bottom half of the dialog shows the summary slots and the components of that sum. In addition, Statistical information is shown at the bottom of the dialog and presents summary information about the highlighted selection. Following is a description and more information on each piece of the dialog.

Select Single or Multiple Object

The user can select to show either a single object or multiple objects (NOTE: currently only a single object has been implemented)

Figure 2-59:



Select Objects

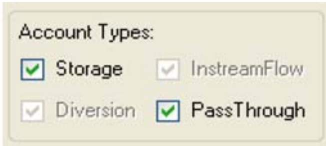
The user can select the object using either a pull-down menu or through the slot selector dialog by clicking on the `Select Object...` button.

Figure 2-60:

Account Types

Accounts can be filtered by type, i.e. Storage, PassThrough, Diversion, or Instream Flow, using the check boxes. The check boxes are enabled based on the types of accounts that reside on the object.

Figure 2-61:



Account Slot Entity

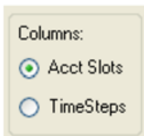
The slot entity (Outflow, Inflow, Storage, etc...) shown in the summary area is selected using the pull-down menu. The most common slot types can be selected using buttons including Inflow, Outflow, Storage, Accrual, Diversion and Return Flow.

Figure 2-62:

Columns

The user can show timesteps as either the column or the rows. If Acct Slots is selected, the accounts are shown across the columns and timesteps go down the rows as shown in the above screenshot. If Timesteps is selected, the timesteps are displayed across the columns and the accounts displayed down the rows in the table. This layout is described in more detail below.

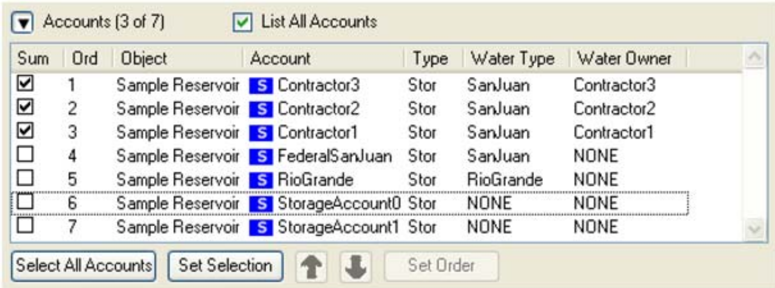
Figure 2-63:



Accounts

The accounts can be ordered and selected for display in the summary area. If the `List All Accounts` toggle is selected, then all accounts will be shown. Otherwise, only the selected accounts are shown. Each column can be sorted by clicking on the column heading. To select multiple accounts, highlight the row and click `Set Selection`. All accounts can be selected using the `Select All Accounts` button. The accounts can be re-ordered by two methods: click on the column heading to sort the accounts by that column, or select an account and click on the up or down arrow to move the accounts. Once the order is defined as desired in the Account Selection area, click on the `Set Order` button to apply the changes. The reorders the slots in the summary area. Click the  button to show/hide the Accounts area.

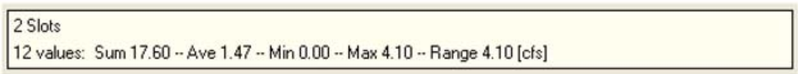
Figure 2-64:



Statistics Information

The following summary statistics are displayed for the highlighted selection: Sum, Average, Minimum, Maximum, and Range.If the slot entity shown is a Flow slot, then the summary area also shows the sum of the selected cells converted to a volume.

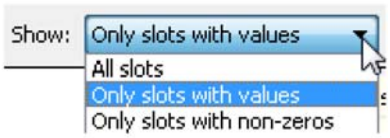
Figure 2-65:



Slot Filters:

The `Show` menu allows you to specify whether to show All Slots, Only Slots with values (non NaNs) or Only slots with non-zero values.

Figure 2-66:



Summary Area

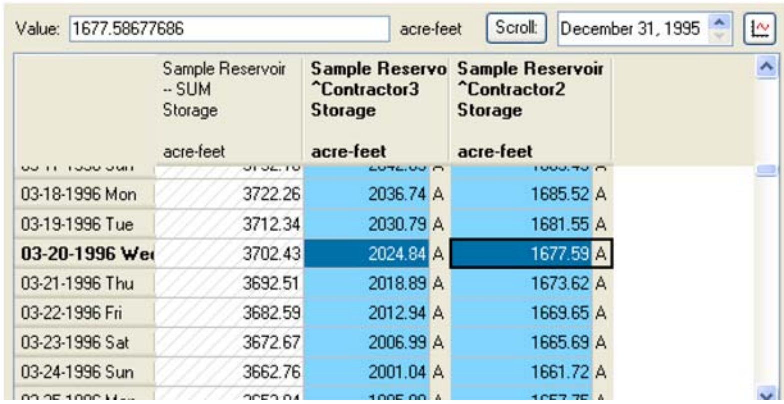
The summary area shows each of the selected slots or timesteps and the total sum for the selections. If appropriate, values in the summary area can be switched from Flow to Volume using the toggle button. Type in a date in the date box to and click `Scroll` to move to that timestep. Use the plot button to plot the selected slot. If one or more slots are selected, the plot button will plot the selected slot for the entire time period. Click the button to hide the Timestep area and the button to show the area if it is hidden. The summary area is laid out differently depending on whether the Slots or the Timesteps are selected to be used as columns. The two options are described below:

Figure 2-67:

Slots For Columns

If the user has selected to show the Slots as Columns, then the dialog looks as follows.

Figure 2-68:

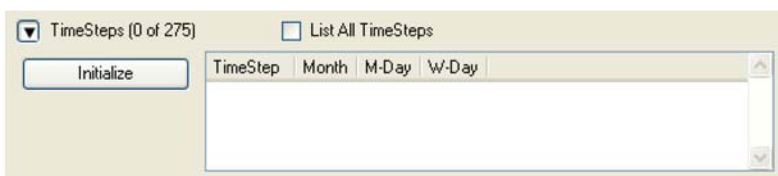


There is also the ability to hide or compress repeated values, a specified value, or NaN's. For more information about Series Display Compression, see "Series Display Compression" in Slots.pdf.

Timesteps for Columns

As described above, the user can select to either show slots or timesteps as columns in the summary area. If the user selects the button to show timesteps as the Columns, the layout changes to add an area where the user can select the timesteps to show. At first, the area is empty like the following screenshot.

Figure 2-69:



The user must click on the `Initialize` button to activate the timesteps. This populates the table with the timesteps as shown in the following screenshot.

Figure 2-70:

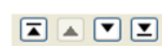
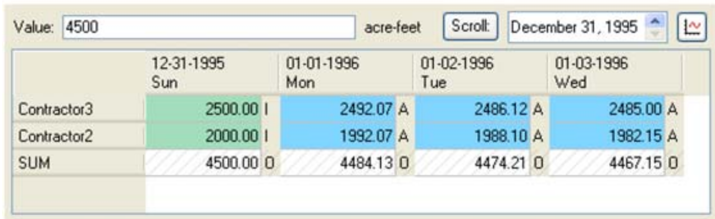
Click on the `List All Timesteps` toggle to show all of the timesteps in the model. Use the arrows  to navigate through the model's timesteps. Select the desired timesteps by checking the box to the left of the timestep. Click the `Set Selection` button to apply your choices and update the summary area. The dialog now is similar to the following screenshot.

Figure 2-71:

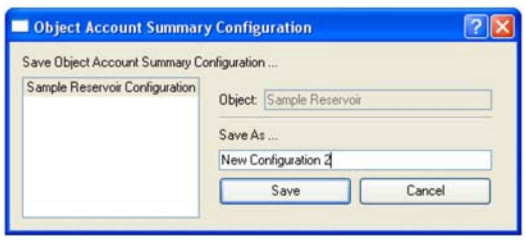


In this layout, the summary information is shown as the bottom row of the dialog. Again, the user can highlight specific portions to see the summary statistics and to initiate a plotting operation.

Save Configuration

The configuration can be saved using the Config ->Save As menu. This opens the following dialog where the user can specify a name for the configuration.

Figure 2-72:



This configuration is saved with the model file. To apply the configuration to an Object Account Summary use the Config menu and select the previously defined configuration name.

Chapter 3
Data Object Exchanges

As an alternative to RiverWare's Exchange utility, (see "Exchange Description" on page 154), some RiverWare users employ a data object based approach to track water exchanged between accounts. In this approach, the user tracks the exchange with rules that reference specifically named slots on data objects. This section describes a utility, called the `Data Object Exchange Builder`, that helps to create this type of exchange by adding slots to the data object and creating supplies on the workspace. The rules must be defined by the user; no further discussion is provided as to what the rules do or how they are written.

This chapter covers the following topics:

Data Object Exchange Overview

The data object exchange builder has capabilities as follows:

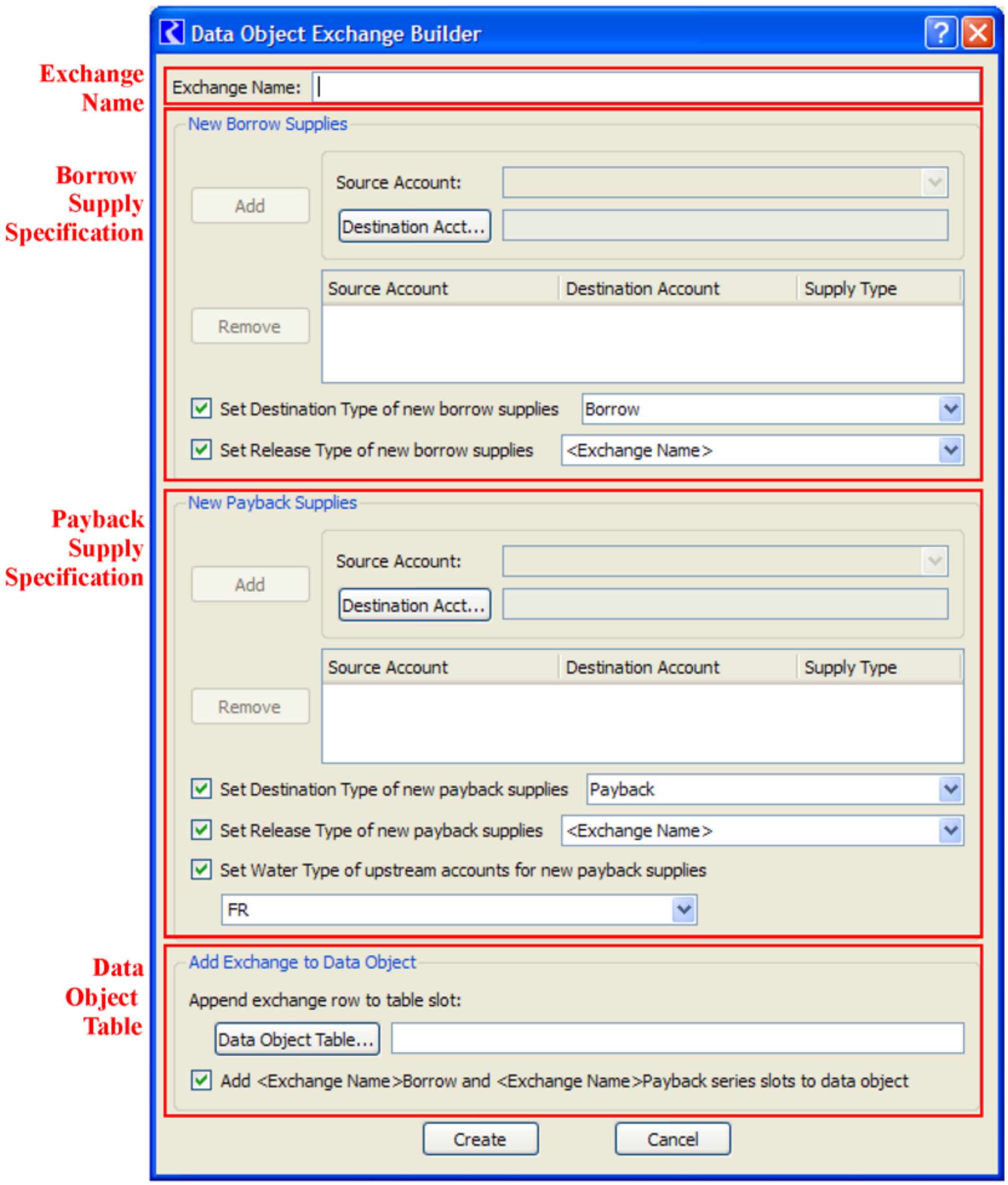
- A row is created in a table on a data object. The row is labeled with the name of the exchange and the various columns in the table represent exchange characteristics.
- Series slots may be created on the data object to keep track of the amount of water exchanged. The slots are named <exchange name>Borrow and <exchange name>Payback.
- Borrow and payback supplies are created between desired accounts to create paths over which the exchange water moves. Optionally, the borrow supplies can be given a Destination Type of “Borrow” and the payback supplies can be given the Destination Type of “Payback”. Both borrow and payback supplies can be given a Release Type that is the name of the exchange.
- The Water Type of the source accounts for payback supplies may be given a type describing the exchange purpose.

Again, these are simply pieces of a Data Object Exchange, that when accessed with the RiverWare Policy Language (RPL), can be used to move water between accounts and track the water that was moved.

Data Object Exchange User Interface

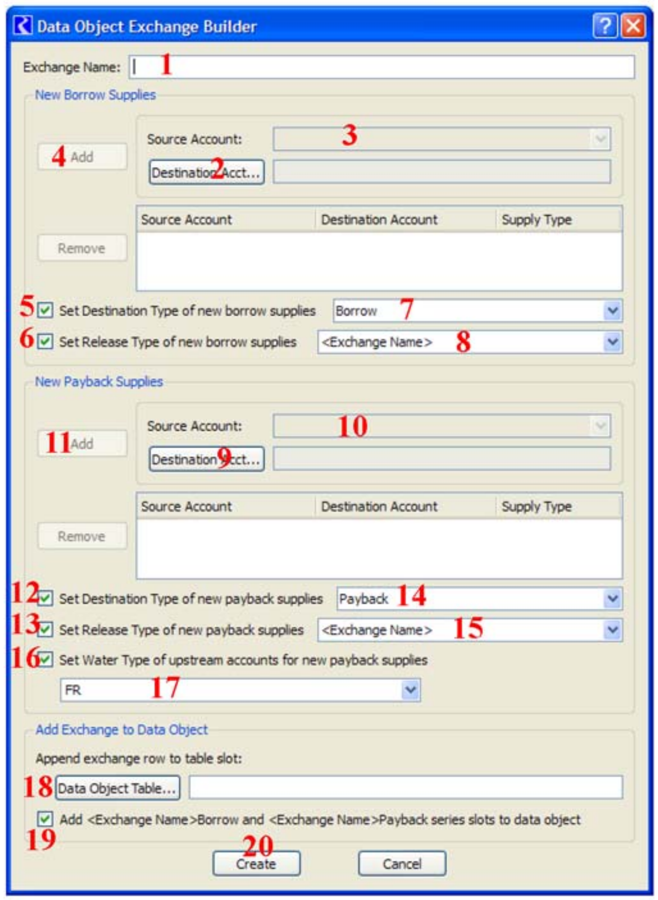
The Data Object Exchange Builder dialog is accessed from the Accounting > Data Object Exchange Builder... menu of the main RiverWare workspace. This dialog is a modal dialog (no other dialogs can be accessed or edited while this dialog is open) and is used to create components for the data object exchange. A screenshot of the dialog follows:

Figure 3-1:



The dialog has four main areas as described in the following sections. When creating a data object exchange, the user typically follows the order as specified on the figure and noted in the text.

Figure 3-2:



Exchange Name

An `Exchange Name` must be entered for the data object exchange. (1) The name will be used as the row label for the exchange's row in the Data Object Table. The name may also be used as a Release Type for the supplies and as part of the name for the Borrow and Exchange series slots created on the data object.

Borrow Supply Specification

In the accounting system, the creation of supplies is always from the destination account (downstream) to the source account (upstream). It is the same in this dialog; first the destination account is selected, then the source account is selected. A `Borrow Supply Specification` is created by clicking the `Destination Acct...` button (2) in the New Borrow Supplies frame of the dialog. This opens an account selector dialog where the destination account for the new supply can be chosen. The `Source Account`**:** combo box (3) is then populated with potential source accounts for the supply based on the destination account that was selected. After selecting a source account, clicking the `Add` button (4) will create an entry for the borrow specification in the borrow list box. If the supply is not valid, a message will be posted to a warning dialog and the supply will not be added to the list. Additional borrow specifications can be created and added to the list by repeating the above steps. At least one borrow supply must be specified to create a data object exchange. A specification can be removed from the list by selecting the row and clicking the Remove button.

The check boxes in the New Borrow Supplies frame optionally allow the user to select a Destination Type (5) and a Release Type (6) to be assigned to borrow supplies when they are created. The Destination Type combo box (7) is populated with all destination types defined in the model plus a type called Borrow if it does not already exist. The Release Type combo box (8) is populated with all release types defined in the model plus a type with the name of the exchange.

Payback Supply Specification

Payback Supply Specifications are created with controls that behave the same as those described above for borrow specifications. (9-11) At least one payback supply must be specified to create a data object exchange.

The New Payback Supplies frame of the dialog also has check boxes for optionally setting the destination type (12) and release type (13) of the new payback supplies. The Destination Type combo box (14) is populated with all the destination types defined in the model plus a type called Payback if it does not already exist. The Release Type combo box (15) is populated with all release types defined in the model plus a type with the name of the exchange. The New Payback Supplies frame also has a check box (16) to optionally set the water type of upstream accounts of the new payback supplies. The Water Type box (17) is populated with all the water types defined in the model, plus this box allows typing in a new water type, if desired.

Data Object Specification

A Data Object Table must be specified for the exchange. Clicking the Data Object Table... button (18) will bring up a slot selector that is restricted to selecting tables on data objects. Once the data object table is chosen, its selection will persist between invocations of the dialog and is saved into the model file so that the selection will persist when saved with the model. When the create button is clicked, a row is inserted or appended after the last named row on the table and is labeled with the exchange name.

The check box (19) in the Add Exchange to Data Object frame allows the optional creation of a borrow and a payback series slot on the data object for the exchange. The slots will be named with the exchange name and the suffix “Borrow” and “Payback”.

Exchange Creation

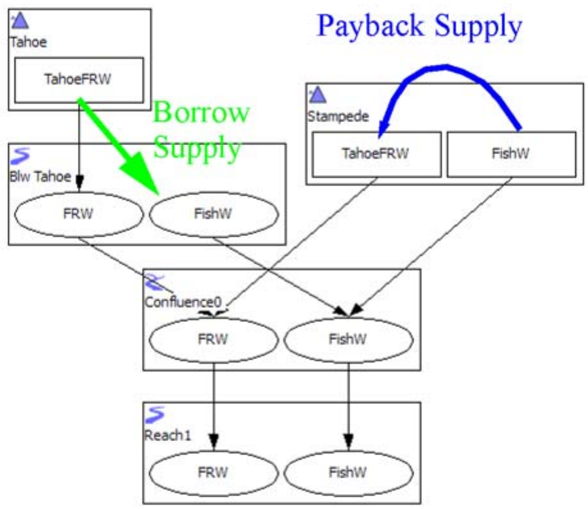
Clicking the Create button (20) will create the specified components of the data object exchange. A warning dialog is generated if required information is missing or if some component cannot be successfully created. A final summary message box is generated listing the components that were successfully created for the exchange. Closing the message box will then dismiss the Data Object Exchange Builder dialog.

Data Object Exchange Example

The following shows an example where a data object exchange (named Article8S) could be created to track water transfers. Water is borrowed from the TahoeFRW account on Tahoe reservoir to be released for fish and wildlife purposes. The water is paid back on Stampede reservoir by a transfer of water from a fish and wildlife account to a

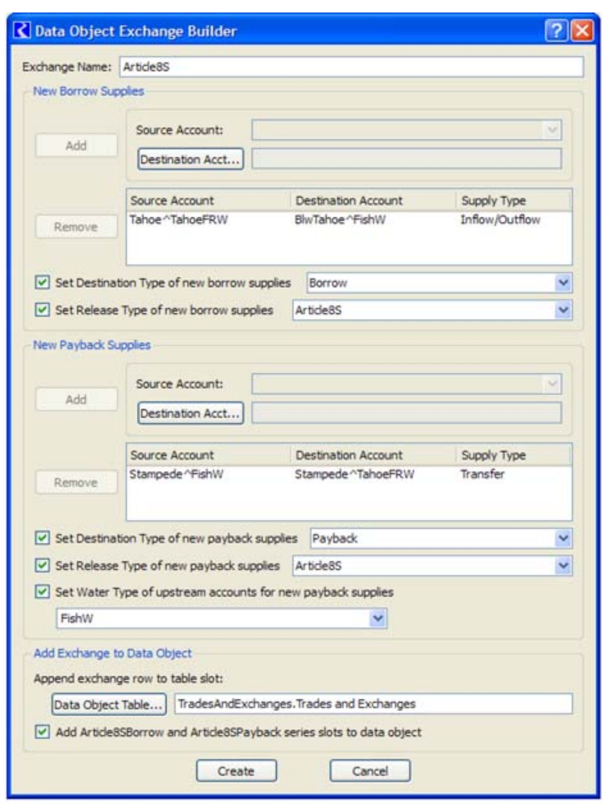
TahoeFRW account. This exchange requires that two new supplies be created as shown by the colored arrows below.

Figure 3-3:



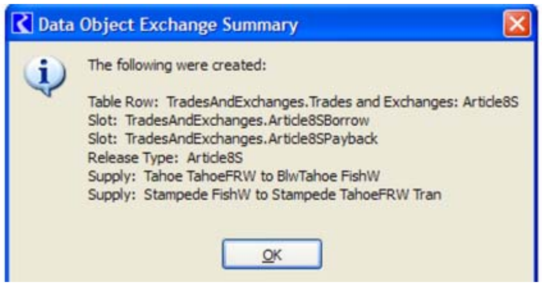
The following screen shot of the Data Object Exchange Builder dialog shows the selections that are made to create a data object exchange for this example.

Figure 3-4:



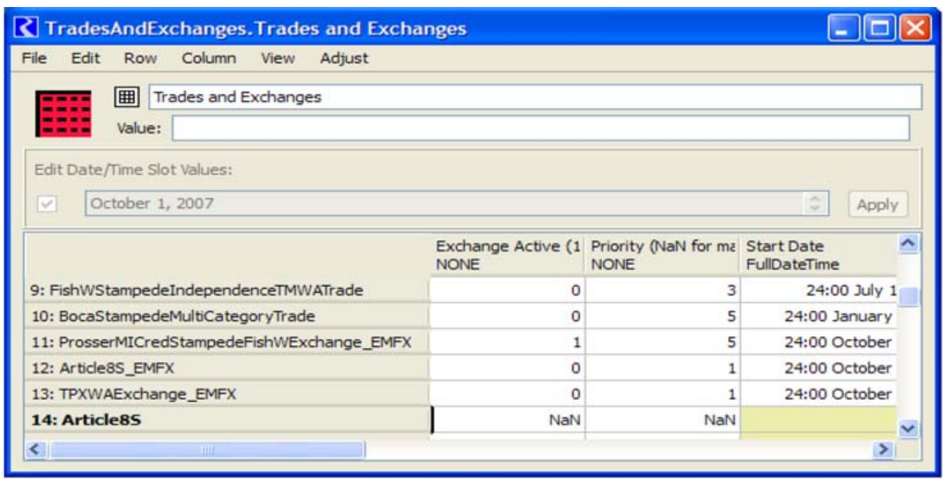
Clicking the `Create` button generates the following summary message of what was created for the exchange.

Figure 3-5:



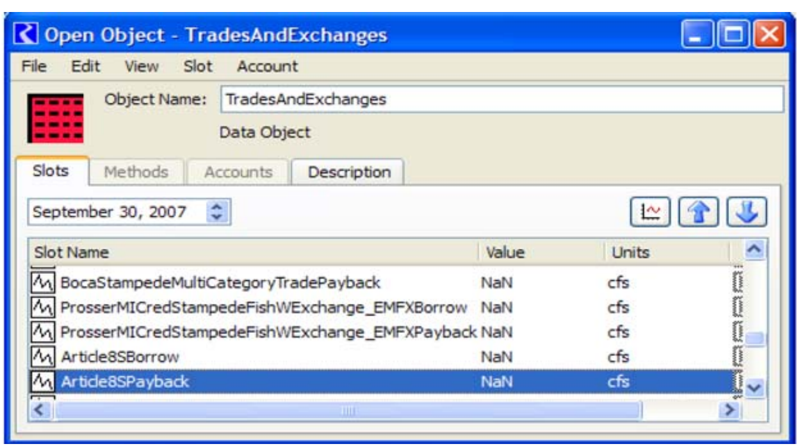
The following screenshot shows the row that was created on the Trades and Exchanges table slot for the new data object exchange. The user would need to fill in the column data to define the characteristics of the exchange.

Figure 3-6:



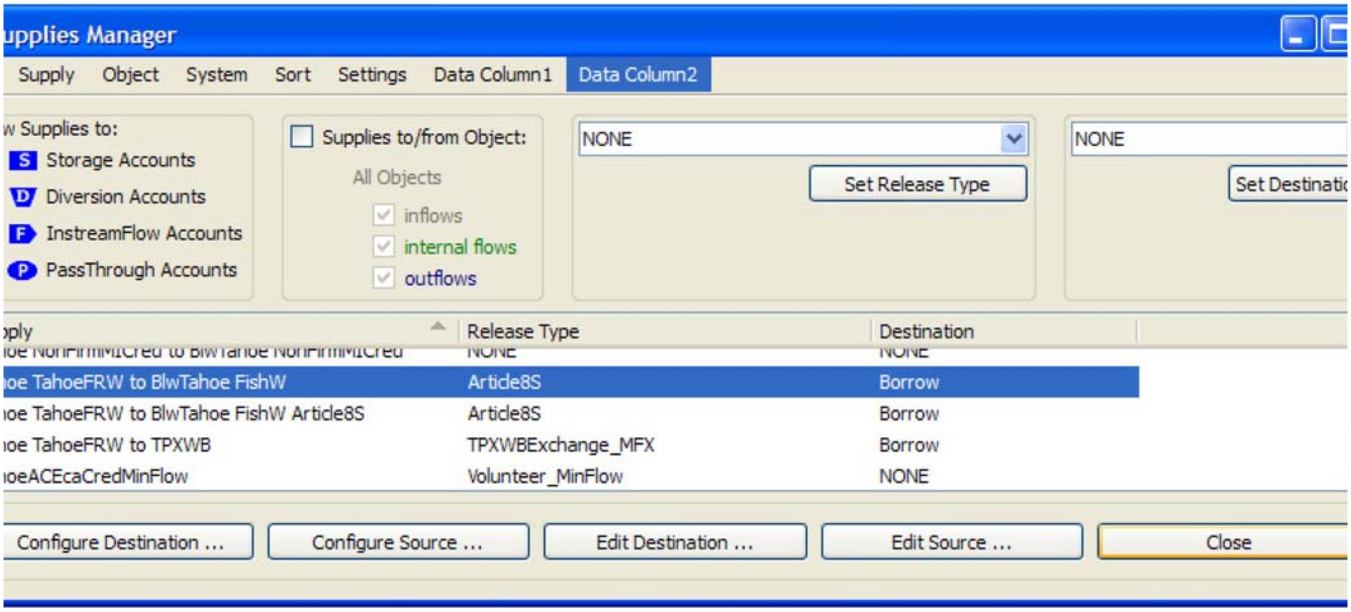
On the TradesAndExchanges data object two new series slots for borrow and payback have been added for the exchange.

Figure 3-7:



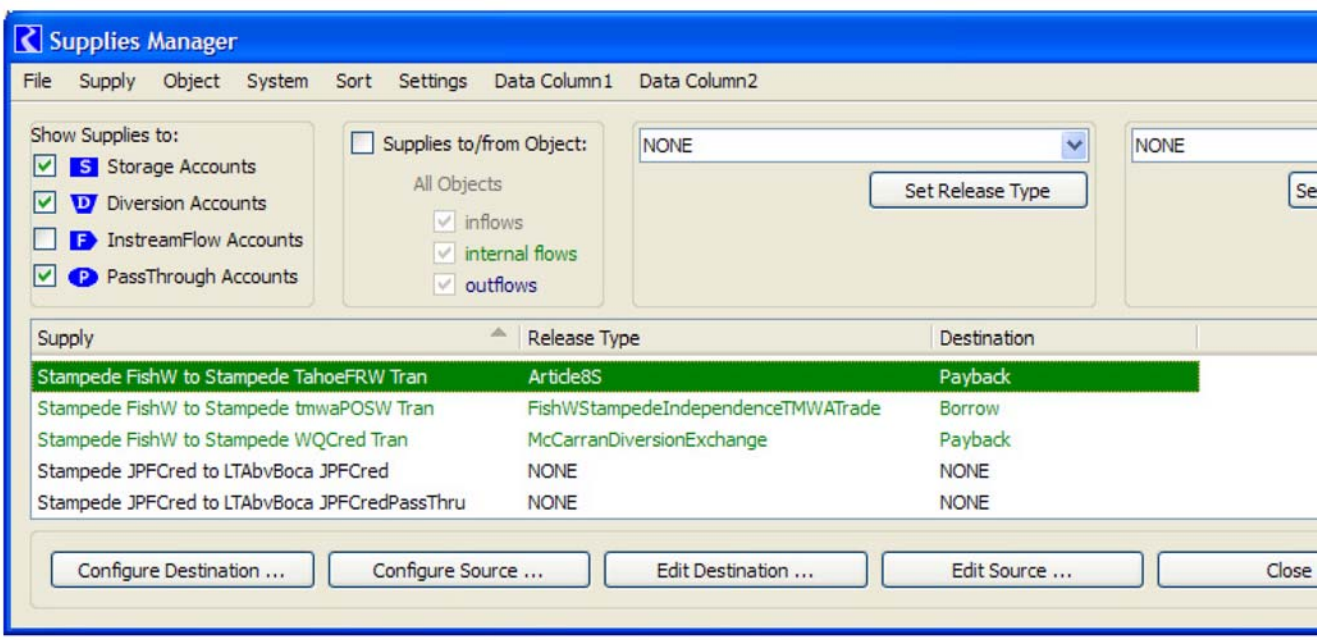
The following screenshot of the Supplies Manager shows the borrow supply that was created for the exchange. Note that per the specifications in the Data Object Exchange Builder dialog, the release type of the new borrow supply was set to Article8S and the destination type was set to Borrow.

Figure 3-8:



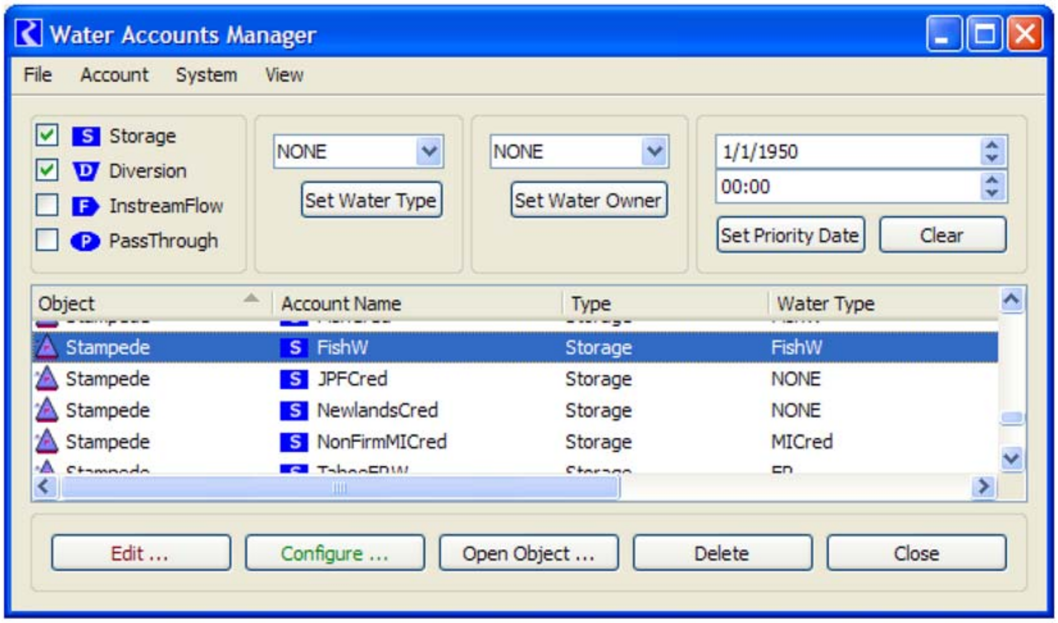
The following screenshot shows the payback supply that was created. Note that per the specifications in the Data Object Exchange Builder dialog, the release type was set to Article8S and the destination type was set to Payback.

Figure 3-9:



The following screenshot of the Accounts Manager dialog shows that the water type for the upstream account of the payback supply has been set to FishW as specified in the Data Object Exchange Builder dialog.

Figure 3-10:



This example has shown how the various pieces related to a data object exchange can be specified and created from the Data Object Exchange Builder.

Chapter 4
Exchanges

An exchange is a RiverWare utility that models a special relationship amongst a set of supplies.

This chapter covers the following topics:

Exchange Description

An exchange is defined by the user to keep track of debts when (paper) water is borrowed from one account and then later returned (if necessary). Borrowed water is represented by a transfer from one account to another through a supply. Repayment of the debt is represented by one or more different supplies.

> Exchange
>
> An exchange is a utility used to track the borrow and payback of water from one account to another.

An exchange in RiverWare only models relationships among entities in the accounting network, In as much as a model uses an accounting network to model ownership and movement of water in the physical network, exchanges can be used to represent physical transfers also.

For example, an exchange can be used to represent:

- A transfer of water: Water must be released from reservoir A due to maintenance concerns and the water is stored in downstream reservoir B. To pay back the exchange, reservoir B pumps water to local irrigators that are in reservoir A’s service area.
- A change of ownership: Water in a reservoir is transferred from owner 1 to owner 2 but no physical release was made. The exchange may or may not be paid back.
- In-lieu of release: Reservoir A is obligated to make a release for downstream diverters. Instead it is negotiated that parallel Reservoir B will make the release and satisfy both the demand and some other purpose like power production. The exchange is paid back by reversing the releases, Reservoir A releases when there is no power demand but minimum flows that would have come out of Reservoir B must be met. These types of exchanges are typically negotiated to try to provide multiple benefits by releasing from the most appropriate reservoir, not necessarily the reservoir that is supposed to release.

Exchange Components

To represent the location and/or quantity of borrowed water, an exchange contains a series slot called either “Borrow” or “Input Borrow”, depending on how the exchange is configured (described in detail below). Regardless of the slot name, this slot represents the amounts of paper water borrowed.

The exchange offers two different ways to monitor the resulting debt (un-paid balance). The first way is required, the second is optional.

First, the exchange must have one or more payback supplies, each of which contributes to the payback, thereby reducing the entire debt. The exchange keeps track of the entire remaining debt (considering all payback sources) in a slot called “Source Balance”, described in detail below. Each payback source may have a loss associated with it. For each payback, the exchange keeps a slot to monitor the debt that payback source would have to pay to resolve the entire remaining debt by itself, taking into account the loss associated with that payback source.

Second, the exchange may (optionally, and in addition to monitoring payback sources) monitor the paper water flowing through a single Payback Destination Supply. No loss may be associated with this supply. The exchange keeps track of the entire remaining debt according to this scheme with a slot called “Destination Balance”. Configuring the exchange to use this monitoring scheme is described in detail below.

Following is a description of the components to the exchange including the Borrow, Payback Source Demand and Payback Destination Supply. The Exchange can be tracked by two separate balance slots, the source balance and the destination balance. The source balance keeps track of the borrowed water and the debt (including a legal loss term) that must be repaid through the Payback Source Demands. Alternatively, the exchange can be tracked using the Destination Balance. This tracks the borrowed water and water that arrives at the Payback Destination. This is a single supply and no loss is allowed.

To make an exchange, the user must specify the supply and give it a value (using either input, rules, object level method) or input the amount of water that is borrowed. This is called the Borrow.

> Borrow
>
> A supply (or user input) representing the water that initiates the exchange.

In the exchange balance, a slot is added called the Borrow (or Input Borrow) slot:

+ **Borrow Supply** or **Input Borrow Supply**

Type:	Series Slot
Units:	volume
Alt Units:	NA
Description:	The supply that represents the water borrowed as part of an exchange. This is typically a reference to a supply in the model. In that case, the exchange displays the slot as “Borrow Supply.” It can also be configured such that the borrow amount is input directly into the exchange balance dialog. In that case, the slot is called “Input Borrow Supply” and is created on the exchange object to store the user inputs.
I/O:	Input or set by a rule
Supply Links:	No

To keep track of the total debt at each timestep based on payback sources, the exchange maintains a slot called *Exchange Source Balance*. To continue the example, the statement you receive from the bank each month shows the balance remaining on your house loan. This is the Source Balance. If you take out a second mortgage to remodel your house (i.e. borrow more), your source balance will increase. If you make payments on the house the source balance decreases. The Source Balance is calculated as:

$$SourceBalance = SourceBalance(t-1) + Borrow - \sum(payback_{(t)} \times lossCoeff_{(of\ this\ payback)})$$

+ **Source Balance**

Type:	Series Slot
Units:	Volume

Alt Units: NA
Description: The Source Balance is the previous Source Balance plus the amount in the Borrow Supply minus the sum of any Payback Source Demands (including loss).
I/O: Exchange Solution
Supply Links: No

An exchange is paid off through paybacks represented by one or more supplies. The various payback supplies are called Payback Source Demands. The nomenclature “demand” can be confusing to many users. Conceptually, a supply is water entering an account (i.e. supplying that account with water); a demand is water exiting an account (i.e. water is demanded from that account). In RiverWare, the term “demand” refers to the receiving/downstream end. “Supply” refers to the supplying/giving/upstream end as well as to the slot that propagates to both ends, or to the combined supply/demand pair. As a result, RiverWare typically refer to all transfers between accounts as supplies. The Payback Source Demand is one of the few places in RiverWare where the transfer is called a demand. Each Payback Source Demand may have a loss associated with it, which is a value computed from a loss coefficient associated with the payback source.

> Payback source Demand
>
> Payback Source Demands (i.e. Supplies) are one or more supplies where an exchange may be considered to be “paid back.” Each Payback Source can have a legal loss associated with it.

For each Payback Source Demand, a slot is added to the exchange that represents the debt that would have to be paid back to fully satisfy the source balance

$$Debt_{(t)} = \frac{srcBalance_{(t)}}{1 - lossCoeff_{(of\ this\ payback)}}$$

+ Debt (for each supply specified)

Type: Series Slot
Units: Volume
Alt Units: na
Description: For each payback source, the exchange maintains a slot Debt to keep track of a debt associated with that payback source. It is the amount that would be necessary in the given supply to pay off the total debt including any loss.
I/O: Exchange Solution
Supply Links: No

To keep track of the total debt at each timestep based on a destination payback supply, the user selects a Payback Destination Supply:

Payback Destination Supply

A single supply where the exchange's payback may be monitored. No loss is associated with this supply.

When the user selects a destination, the Destination Balance slot is added to the exchange and is calculated as:

$$DestinationBalance = DestinationBalance(t-1) + BorrowSupply \\ - PaybackDestinationSupply$$

+ Destination Balance

Type: Series Slot
Units: Volume
Alt Units: na
Description: The Destination Balance is the previous Destination Balance plus the value in the Borrow Supply minus the value in the Payback Destination Supply.
I/O: Exchange Solution
Supply Links: No

An exchange will re-solve whenever a new value is set on its borrow, payback source(s), or destination payback supplies. The borrow and payback source(s) values may be set via user input or rules. Borrow values may be set on a supply or as a user input on a slot on the exchange object (user input demand). The effects of each new value are listed below:

- A value on a supply or user input slot registered as a Borrow will increase the debt on that exchange; Source Balance, Destination Balance, and the Debt (for each payback) will increase.
- Releases from a supply registered as a Payback Source Demand on an exchange will reduce the debt on the exchange; Source Balance and the Debt at each payback will decrease.
- Releases arriving at a Destination will reduce the Destination Payback Balance. This is NOT the same as reducing the debt.

Accessing Exchanges through RPL

The debt of a given exchange, if paid through a certain payback source, is the value of the debt slot on that payback. Two RPL functions enable users to access the debts:

- GetPaybackDebt (STRING SupplyName, DATETIME date)

  The SupplyName argument is a string representing the Source Payback Demand. The date argument is the date at which to get the payback debt. The function gets the value of the debt associated with the given Payback Source Demand at the given timestep.
- GetObjectDebt (OBJECT object, DATETIME date)

This function evaluates to the sum of the debts to all accounting exchanges which may be paid by supplies on the given object at the given timestep. If there are no exchange paybacks on the given object, the debt is zero.

Exchange Examples

Following are two examples that illustrate exchanges and how they are defined in RiverWare. The first example is a toy problem but explores the configuration of the supplies. The second example is more realistic and presents alternative configurations for the exchange.

Example 1

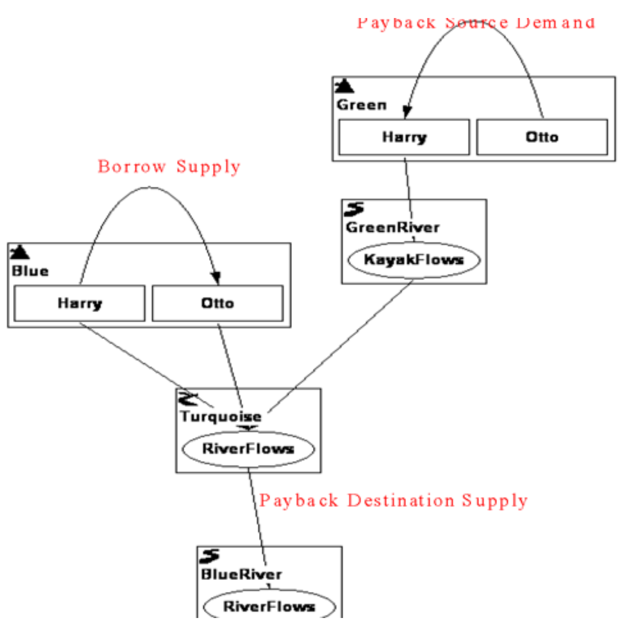
Two water owners, Harry and Otto own water in two different reservoirs, Green Lake and Blue Lake. On each of the reservoirs are accounts Harry and Otto. At Blue lake, Otto owns a dock for fishing and wishes to keep the reservoir as high as possible during the summer. As a result, he has agreed to borrow water from Harry during the summer for fishing. He will pay back any water borrowed by transferring water at Green lake from Otto to Harry. Harry loves this idea because he is an expert kayaker and now controls more water in Green Lake which he can release directly into his favorite kayaking reach, GreenRiver. Let’s set up this exchange:

We establish a supply transferring from Blue^Harry to Blue^Otto to represent Otto borrowing water from Harry at Blue Lake. This supply will be called Borrow. Otto returns water to Harry at Green Lake; this is represented by a supply called Payback, which will be a transfer from Green^Otto to Green^Harry. All the water that flows through Borrow at time t increases Otto’s debt at time t, and all that flowing through Payback decreases the debt at time t. In the exchange manager, Borrow is the Borrow Supply and Payback is the one and only Payback Source Demand.

Returning to the example above, Otto decides that Kayaking isn’t for him but instead wants to innertube on the calm BlueRiver downstream of both reservoirs. We still wish to keep track of the debt at Blue Lake, but perhaps he does not want the debt to be considered repaid until the water reaches the Blue River downstream of both lakes. To model this, we choose “Turquoise RiverFlows to BlueRiver RiverFlows” as the Payback Destination Supply. Water flowing through this supply represents the total water paid back, and the balance at time t is maintained in the exchange’s slot Destination Balance. In this way, we see that any water released out of either Blue or Green would repay the exchange.

The diagram below shows the layout of this accounting system and the appropriate supplies.

Figure 4-1:



Example 2

In this example, we first describe the physical conditions, background, and policy that motivate this exchange. Then we describe how to define this exchange in RiverWare.

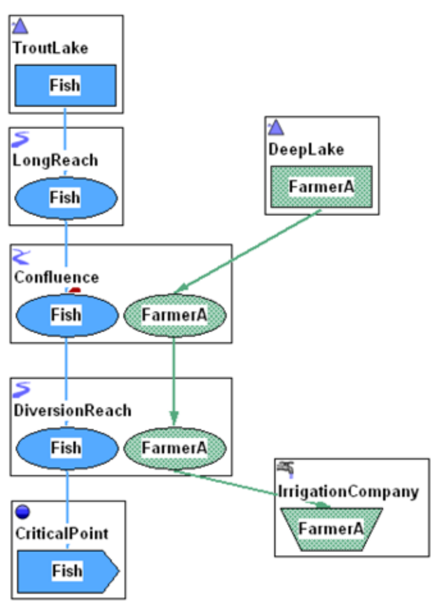
The following diagram shows a layout of a basin consisting of two reservoirs. TroutLake was built specifically to make releases to meet Fish Targets at the CriticalPoint control point for endangered species. All of the water in TroutLake is stored in the Fish storage account. A chain of passthrough accounts, also named Fish, was created to move the water from the Fish storage account to the Fish instream flow account on the CriticalPoint control point.

DeepLake was built and is owned by an irrigation company. Water stored in this reservoir is released to meet farmer's demands at the IrrigationCompany water user. To represent this, a storage account name FarmerA was created in DeepLake and passthrough accounts were created to move the water from that account to the FarmerA account on the IrrigationCompany water user. This example shows a small sample of the accounts in this basin. Likely there are other farmer accounts on DeepLake and on the IrrigationCompany. In addition, there is likely policy in the basin to simulate other objectives like flood control. This example simplifies the policy but it is good to be aware that this exchange must fit into the other operating objectives.

TroutLake is a relatively small compared to DeepLake. During the hot summer months, TroutLake often does not have storage or release capacity to entirely meet the required fish flows at the CriticalPoint. To maintain adequate habitat for the endangered fish species, an exchange agreement was negotiated to between the Fish and FarmerA accounts to provide more water for the fish. The agreement states that if TroutLake is low and cannot meet Fish Targets, FarmerA will release water from DeepLake that will be allowed to flow to the CriticalPoint. The balance of the exchange will be repaid, including a legal loss of 5%, by releasing additional water from the Fish account at TroutLake that can then be diverted to FarmerA. Now we will define how this exchange will be configured in RiverWare. First, let's create and define a borrow supply. The borrow supply must allow FarmerA water to release to the Fish account. This supply goes from DeepLake^FarmerA.Outflow to Confluence^Fish.Inflow. This is the borrow supply.

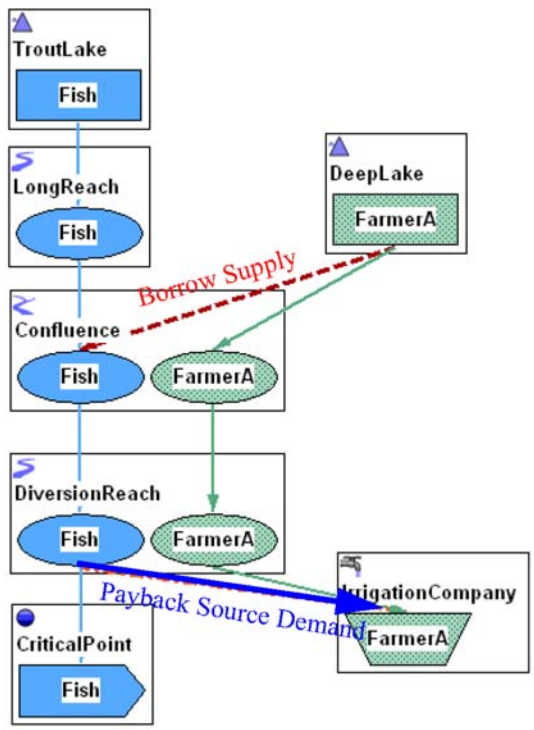
We will go through two cases. One with just a source payback and the other with a source payback and destination payback.

Figure 4-2:



In the first configuration, the exchange is paid back by releasing additional water from the Fish account that can be diverted to FarmerA. We must create a supply from DiversionReach^Fish.Diversion to IrrigationCompany^FarmerA.Diversion. This is the payback supply. The following screenshot shows this connection. In this way, water can be borrowed from FarmerA and allowed to flow to the Fish instream flow account. It can also be paid back by releasing water from TroutLake through the series of Fish accounts, then diverted to the FarmerA account on the water user. Rules or user inputs will be used to set values on these exchanges. In this configuration, water is paid back only when it is diverted to FarmerA. If there is a legal loss, this supply would be defined as the Payback Source Demand. If there is no loss, this could be defined as a Destination Payback. Until water reaches this diversion supply, the paper water still belongs to the Fish account and will incur any Fish losses in LongReach.

Figure 4-3:

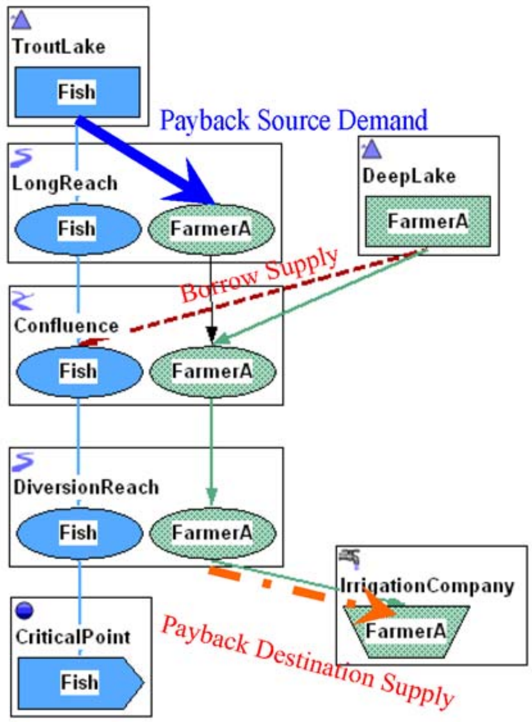


In the second configuration, the Borrow supply remains the same but we will create a separate FarmerA passthrough account on the LongReach and supplies connecting it to the other FarmerA passthrough accounts. The supply connecting TroutLake^Fish.Outflow to LongReach^FarmerA.Inflow is the Payback Source Demand. A second diversion supply will then be created from DiversionReach^FarmerA.Diversion to IrrigationCompany^FarmerA.Diversion. Now there are two supplies connecting these two accounts. This new supply is the Payback Destination Supply. The following diagram shows these new accounts and supplies.

In this configuration, once water is released from the Fish account to repay the exchange, it becomes FarmerA water and may or may not incur loss (depending on policy) in the Long Reach. By defining a Legal Loss, on this Payback Source Demand, we can specify that more water is required to repay the exchange than was borrowed, i.e legal loss was incurred. We can also define the Payback Destination Supply and then the exchange will track the Destination Balance in addition to the debt remaining from Payback Sources. Rules can then look at the Source

Balance, Payback Debt, and Destination Balance to determine how much water to release to repay the debt

Figure 4-4:



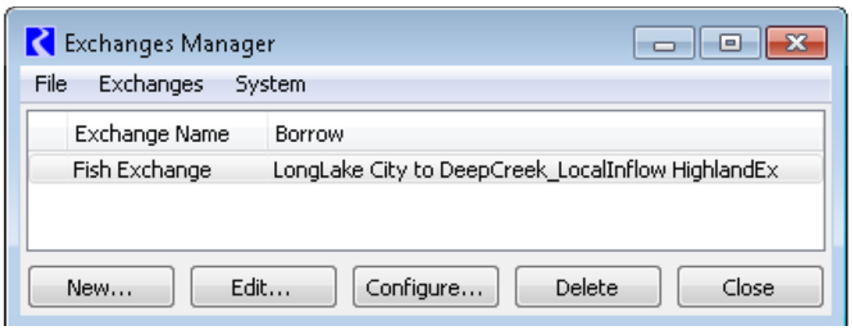
Exchange User Interface

Following is a description of the exchange user interface. <Link to sub-headings needed.>

Exchanges Manager

Exchanges are created through the Exchanges Manager dialog. This is accessed from the workspace through the Accounting > Exchanges Manager. It can also be accessed from other locations such as the accounts manager and supplies manager. The following screenshot shows the Exchange Manager with one exchange defined. It lists each exchange and the Borrow supply (or user input) associated with that exchange. Items in this list can be sorted by clicking on either column heading.

Figure 4-5:

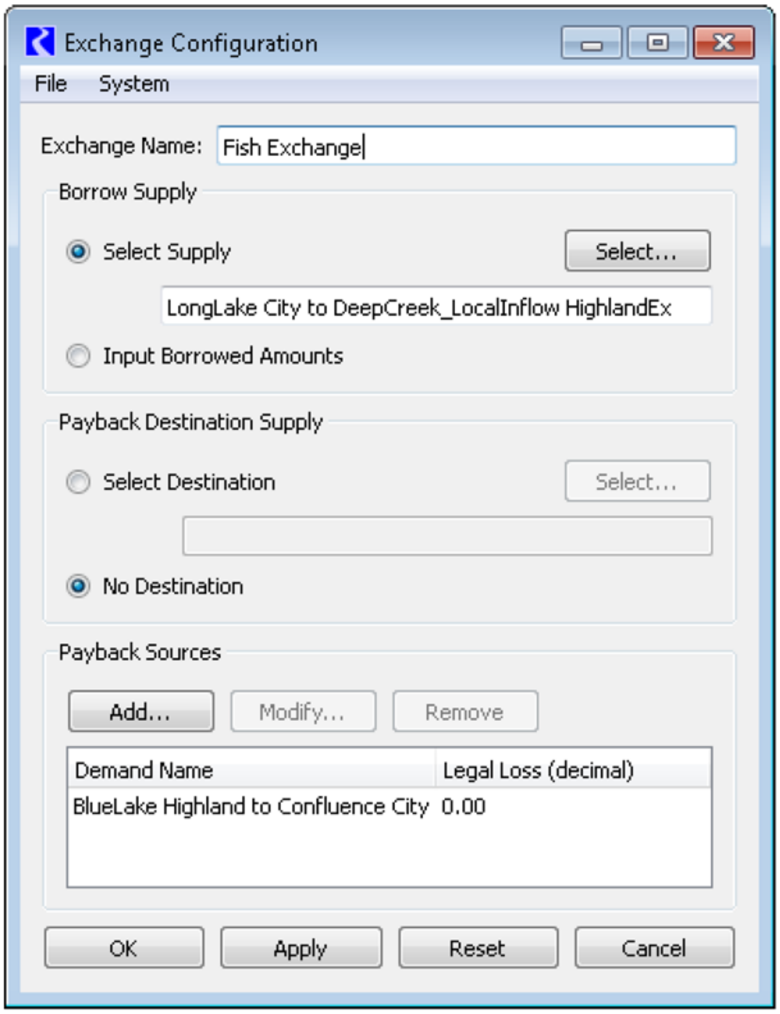


To edit data (or view) for an existing exchange, highlight the exchange and click Edit. This brings up the Exchange Balance dialog. We will explore this later. To create a new exchange click New and a blank Exchange Configuration dialog will open. To modify an existing exchange click Configure to open the Exchange Configuration dialog.

Exchange Configuration dialog

The Exchange Configuration consists of four areas: Borrow Supply, Payback Destination Supply, Payback Sources and Units. On the first line, the user can enter an Exchange Name in the text field.

Figure 4-6:



The `Borrow Supply` can be specified using the radio buttons to select either that the user will use the `Select` button or that the user will `Input Borrowed Amounts`. If the `Select` option is used, the text field becomes active. The user can either type in the supply name (not recommended) or click the `Select` button to select the supply using the standard selection mechanism.

The user can then configure the `Payback Destination Supply`. To reiterate, either zero or one `Payback Destination Supply` is allowed. The user can select whether to `Select Destination` or have `No Destination`. Again, if the user chooses to `Select Destination`, the user can either type in the supply name (not recommended) or click the `Select` button to select the supply using the standard selection mechanism.

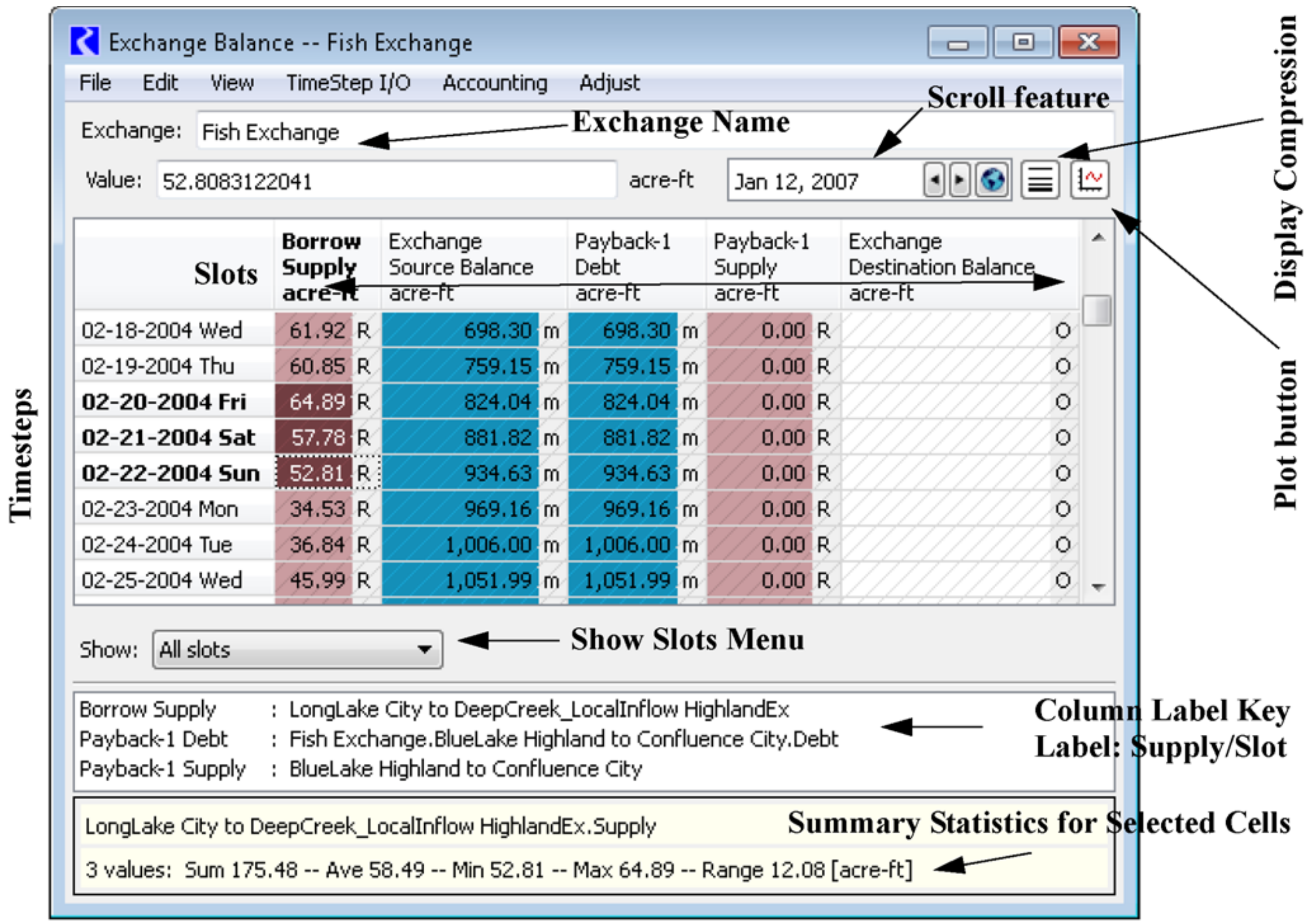
The third area, `Payback Sources` allows the user to specify the `Payback Sources` which consist of one or more supplies and legal loss associated with each supply. To add a `Payback Source`, click the `Add` button and select the supply. Then enter the `Legal Loss` using a click-pause-click action to make the text field editable. It will default to 0.0 if not input. To replace (re-select) any of the existing supplies, first select the supply from the list, then click `Modify` to select a new supply or type in a new legal loss. Use the `Remove` button to delete the selected supply from the exchange.

Click OK or Apply to confirm the changes.

Exchange Balance dialog

The Exchange Balance dialog shows the slots associated with the exchange. Depending on the exchange's configuration, different slots will be shown. The following screenshot shows the Exchange Balance dialog with defined Borrow Supplies, Destination Payback Supplies, and Source Balance, Destination Balance, the Payback Debt, and the Payback Supply.

Figure 4-7:



Slots	Borrow Supply acre-ft		Exchange Source Balance acre-ft		Payback-1 Debt acre-ft		Payback-1 Supply acre-ft		Exchange Destination Balance acre-ft	
02-18-2004 Wed	61.92	R	698.30	m	698.30	m	0.00	R		O
02-19-2004 Thu	60.85	R	759.15	m	759.15	m	0.00	R		O
02-20-2004 Fri	64.89	R	824.04	m	824.04	m	0.00	R		O
02-21-2004 Sat	57.78	R	881.82	m	881.82	m	0.00	R		O
02-22-2004 Sun	52.81	R	934.63	m	934.63	m	0.00	R		O
02-23-2004 Mon	34.53	R	969.16	m	969.16	m	0.00	R		O
02-24-2004 Tue	36.84	R	1,006.00	m	1,006.00	m	0.00	R		O
02-25-2004 Wed	45.99	R	1,051.99	m	1,051.99	m	0.00	R		O

Each column represents a series slot. Those labeled as "Supply" reference a supply in the model. Those labeled as "Balance" or "Debt" are the exchange specific slots. The column labels provide a reference to the actual supplies. The key in the lower part of the window displays the full name of the supply. This allows the columns to remain relatively narrow but still have descriptive names associated with each component. The dialog has similar features to other accounting dialogs including the scroll button, plot button, and "Show Slots" menu. There is also the ability to hide or compress repeated values, a specified value, or NaNs. For more information about Series Display Compression, see "Series Display Compression" in Slots.pdf. There is also the summary area where statistical information can be obtained for the highlighted selection.

In this screenshot, we see that when water goes through the Borrow Supply, the Exchange Source Balance increases as does the Exchange Destination Balance. The Payback Debt for each payback source also increases which includes the legal loss term. The debt is always larger than equal to the source balance. As water passes through the Destination Supply, the Destination Balance decreases. As water passes through the Payback Source Demand (shown as Payback-1 in the figure), the Source Balance and Debt decrease.

Typically the values in the Exchange Balance are read-only; any edits to input data must be made directly on the borrow or payback supplies. The Exchange Balance is a view of the data in those supplies. Also, any of the Balance or Debt columns are always read-only as they are outputs from the solution of the exchange equations.

The one exception to the read-only constraint is when the user has configured the exchange to “Input Borrowed Amounts” in the exchange configuration. Then, the borrow supply lives on the exchange, is labeled “Input Borrow” and is directly editable from the Exchange Balance dialog. Note, when first created, the Input Borrow is empty, so the user must click the “Show All” menu to show this slot. Also remember that any Input Borrow amounts are always entered as a volume.

Chapter 5
Water Right Allocation

This document describes the use of a new rule function in RiverWare that allocates water in the accounting network to water rights (accounts) according to priorities given by the priority date on the accounts. Earlier dates have higher priority than later dates i.e., those with an earlier priority dates have first right to the use the available water over those with later priority dates. This function, along with the methods on accounts and objects, and new linking configurations, support modeling water allocation under the doctrine of Prior Appropriation.

Previously, users have implemented a water rights allocation algorithm using logic created in rules. Although this problem can be solved with rules, the solution is sufficiently complex that the resulting ruleset is large, difficult to maintain, and the resulting model runs are inherently slow for a moderately sized basin. To address these modeling issues, CADSWES has developed a rule function that calls a solver and returns the correct allocations.

First, we present the assumptions/requirements considered in the design and a general description of the solver. Next we describe how to set up a model and rules to use the solver. Finally, the detailed solution algorithm and additional modeling recommendations are presented.

This chapter covers the following topics:

Requirements

The solver's purpose is to determine prioritized water allocations according to the doctrine of ***Prior Appropriation***. In particular, the solver meets the following requirements:

- Three distinct types of water rights can be represented in RiverWare by "legal" accounts, and can receive priority allocations:
  - The right to divert from a stream, represented in RiverWare by a Diversion Account.
  - The right to store water in a reservoir, represented in RiverWare by a Storage Account.
  - The right to minimum in-stream flow, represented in RiverWare by an Instream Flow Account.
- Water rights (legal accounts) that receive priority allocation have a unique priority date, i.e., no two water rights may have the same priority date.
- A right's allowed quantity of water may be described in various ways on legal documents or in practice; the model capabilities must be flexible enough to represent these, and to be extensible to add representations later that are not initially provided.
- Allocation to a junior right may not "short" the available water to a senior right.
- Some rights are subject to ***physical*** constraints such as the physical capacity of a diversion structure or the size of a reservoir pool.
- Some rights are subject to ***legal*** constraints such as accrual-based maxima. Some legal constraints are defined in terms of the state of the network after senior rights have been satisfied, such as minimum bypass requirements that are expressed as a function of the flow in the river.
- The effects of an upstream allocation on a downstream senior right must take into account lags and losses as water moves downstream.
- The solver is intended to allocate only ***Allocatable Flow.*** Hence, it does not solve for any diversions from storage accounts, or releases from storage accounts to meet downstream needs. Such allocations are expected to be addressed with other rules and are not priority-date driven.
- The solver can be executed more than once during a timestep, with different sets of instream flows being allowed to call water from junior upstream rights.

Modeling Approach

To meet these requirements, RiverWare developers designed a rule function that calls a network water rights allocation solver, a new type of legal account to model instream flow rights, some new linking configurations for accounts, and methods on accounts to support the network allocation solver. The following is a summary of the modeling approach.

The water rights solver is designed to work in a rulebased simulation model with accounting. The model must have an underlying accounting network as described in the following sections and the rule set must make calls to the predefined SolveWaterRights() function or, if the accounts have lag times, SolveWaterRightsWithLags(). In particular, the model should consist of:

- Simulation objects with the appropriate methods selected to describe the physical network.
- Legal accounts, i.e., Storage Accounts, Diversion Accounts and Instream Flow Accounts, with priority dates on the simulation objects to represent the prioritized water rights.
- An “allocatable flow supply chain,” modeled by creation of passthrough accounts on the river/reservoir objects (including Reaches, Confluences, Gages, etc.), and the linking together of these passthrough accounts.
- Supplies that link the rights-holding accounts to the allocatable flow supply chain’s passthrough accounts to represent the appropriation points from allocatable flow.
- A computational subbasin that contains the objects involved in the allocation, including all objects with accounts in the allocatable flow supply chain and objects with rights-holding accounts. The computational subbasin must have the appropriate methods selected to allow execution of the water rights solver.
- A result that includes a rule that calls the predefined function SolveWaterRights() or SolveWaterRightsWithLags().

Each of these components is described in greater detail in the following sections.

Simulation Objects and Accounts

Allocations can be computed by the solver for the following configurations:

- A diversion right modeled by a Diversion Account on a Water User or AggDiversionSite Object.
- A storage right on a reservoir modeled by a Storage Account on a Reservoir Object. The reservoir may be in-stream, in which case it contains a passthrough account representing the flow of non-stored native water through the reservoir. It may be off-stream, in which case it receives water from a Reach Object through a Diversion Object.
- An instream flow right modeled by an Instream Flow Account on a Control Point Object; the Control Point sits between Reaches, Confluences, Stream Gages and other object representing the stream.

These water rights are described in various ways in legal documents, so each account with water rights expresses its entitlements through method selections allowing for a variety of ways to express the right’s requested water.

In general, each right makes an initial request based on the legal description of the right. Next, the state of the system at the time of allocation might restrict the initial request by either ***physical constraints*** or ***legal***

constraints to produce a net Allocation Request. The difference between the Initial Request and the amount allocated is the shortage, which represents the portion of the demand that is not met by the allocations from native water. Depending on the method selections on the account, the account may have slots Initial Request and Shortage, which rule sets can use to compute reservoir releases or transfers from other sources or water to meet remaining demand. "Creating a Model" on page 174 describes how to represent each kind of right.

Allocatable Flow Supply Chain

Based on the doctrine of Prior Appropriation, only "allocatable flow" is allocated to the water rights holder. Water already stored in reservoirs or released from those reservoirs is not considered allocatable flow. In this document, water that has been stored and is managed as releases from storage is called "Project" water. To track the allocatable flow through the simulation objects, the model should be configured to have one chain of accounts and supplies (links between the accounts) that is continuous through the portion of the simulation model for which water rights solver needs to be applied. These passthrough accounts are exclusively labeled by a water type that identifies them as the allocatable flow "supply chain," from which appropriations are to be made.

An example of a allocatable flow supply chain is shown in blue in the figure below. Each object has a passthrough account called "Allocatable Flow" which is linked via supplies to other passthrough accounts of the same name.

Linking Rights-Holding Accounts to Allocatable Flow

Allocations can be computed by the water rights solver for the following linking configurations between the allocatable flow supply chain and the water rights accounts:

- A Diversion Account on a Water User or AggDiversionSite Object is linked to the allocatable flow supply chain passthrough account on a Reach via a link is of type Diversion/Return Flow.
- A Storage Account on a Reservoir Object. Two configurations are recognized:
    - For an in-stream reservoir a Storage Account with the storage right is supplied by the passthrough account of the allocatable flow supply chain on the same reservoir via a supply of type Transfer.
    - For an off-stream reservoir, the Storage Account is supplied by a passthrough account on a Diversion Object via a supply of type Inflow/Outflow, which is in turn supplied by a passthrough account on the linked Reach via supply type Diversion/Return Flow.
- An Instream Flow Account on a Control Point Object does not need to be explicitly linked to the allocatable flow passthrough account because the ***Flow*** slot on the Instream Flow Account adds up the ***Inflow*** slots of all the passthrough accounts on the Objects.

Figure5-1 below shows an example of each of these water rights linked to a allocatable flow supply chain.

Figure 5-1:

Figure: 1. Accounting layout showing Allocatable Flow Supply Chain (Blue) and project water (red and green)

A Computational Subbasin that Contains the Objects

The water rights allocation solver refers to a computational subbasin that specifies the objects that contain accounts involved in the solution. The computational subbasin also has slots and methods that are used to control the rule function that uses the subbasin. The method categories pertinent to water rights accounting allow a water rights allocation solver to be used with the subbasin and allow the subbasin to disaggregate annual demands into per-timestep demands. The subbasin is described in more detail in online help document Objects, Section on Computational Subbasin.

At the beginning of the run, if the subbasin is enabled, the subbasin performs checks for each selected method in all categories. For the Prior Appropriation method in the Water Rights Allocation category, the subbasin creates a set of tables describing the topology of the subbasin. It locates all water accounts with priority dates, sorts them, and assigns a value to the account's Temp Priority slot: 0 for the most senior account, and increasing according to later priority dates.

The subbasin records, for each prioritized account, all its supply chains. For each of its supply chains (a chain whose head is the appropriation point for the prioritized account), the subbasin assigns a chain identifier. The tables also record all downstream seniors, reachable from each supply chain. The solver also creates bitmaps to determine if a given account is in a given supply chain. These tables are used during the run by the solver. If a run is paused and resumed, it is easy to see that any topological changes in the model during the pause will render these tables obsolete and can cause the solver to fail in unpredictable ways. In other words, do not make any such changes during a run!

The subbasin also checks for some common modeling errors and reports errors that it finds.

The Rule Set

The ruleset must consist of the following:

The SolveWaterRights() and SolveWaterRightsWithLags() Rules Functions

The water rights allocation solver is executed by calling the predefined function SolveWaterRights() or SolveWaterRightsWithLags() from a rule. This function is called with the following arguments:

1. The name of the computational subbasin that identifies the set of objects (and contained accounts) on which appropriations will be made by this solver
2. A water type identifying the allocatable flow supply chain from which allocations will be made by this solver
3. A date that controls the behavior of the instream flow rights during the solution computation

The SolveWaterRights() predefined rule function returns a list of {slot name, value} pairs, which the rule uses to place the value in the appropriate slot. The function executes the methods on the specified computational subbasin which perform the allocation and return values to the rule. As with other predefined functions, no values in the model are set or modified during the function call. Values are only set at the uppermost level of the rule. For more information about this function, see "SolveWaterRights and SolveWaterRightsWithLags" in RPLPredefinedFunctions.pdf.

The SolveWaterRightsWithLags() predefined rule function works much like SolveWaterRights(), but is used when the subbasin passthrough accounts contain lags. It returns a list of {slot name, date-time, value} triplets, which the rule uses to place the value in the appropriate slot at the appropriate timestep. The timestep given will reflect the Local Timestep Offset of the account on which the slot resides. It is some number of timesteps after the current rules-controller timestep, and reflects the relationships of the account to other accounts in the subbasin based on their respective cumulative lag times to the end of the subbasin.

Overall Rules Order

This section provides an overview of the approach used to allocate water and simulate the network. The model proceeds as follows assuming instream flows are modeled:

- Initialize the accounting system so that the legal entitlements are known or can be determined at run time. This means that all water rights accounts must have an initial request from which their demands can be computed, or, in the case of storage rights, their underlying reservoirs have methods selected to define the bounds of a Conservation Pool, or, in the case of instream flow rights, their underlying control points have a method selected to define a reference level from which their initial requests can be computed.

- Provide the local inflows to the accounting system so that the allocatable flow is known. This is done by rules or Object Level Accounting methods that transfer values from Hydrologic Inflow or Local Inflow slots on the Simulation Object to the Slot Inflow slot on the Allocatable Flow passthrough accounts. If Object Level Accounting Methods are used, they should be configured to execute ("Reconciliation" on page 75) at Beg of Run, Beg of Timestep, or Beg of Timestep Once. Typically, the `Copy Slot to Slot Inflows` method that is described in "Copy Slot to Slot Inflows" can be used.
- Invoke the solver on the allocatable flow supply chain by calling the predefined function SolveWaterRights() or SolveWaterRightsWithLags() with a very early controlling date, i.e., earlier than the earliest Instream Flow right. (if modeling instream flows) to allocate allocatable flow without regard to instream flow rights (they cannot make calls). The quantity of allocatable flow that is available to each Instream Flow account is saved in the slot ***Available Allocatable Flow*** so it can be used by one or more later invocations of the solver that allow these rights to make calls.
- Execute other rules to operate reservoir Storage Accounts to meet downstream unmet demands that could not be met by allocatable flow.
- For each Instream Flow water right, from most senior to most junior, do the following:
- Call SolveWaterRights() or SolveWaterRightsWithLags() predefined function to re-solve, with a controlling date at or later than the current Instream Flow right, but before the next one. The solver will attempt to satisfy demands of each Instream Flow Account whose priority date is older than or equal to the controlling date. These Instream Flow demands are satisfied by both the allocatable flow available at the priority date, and also flow that passes through the object as a result of reservoir storage releases. In this invocation, allocation to upstream juniors are not allowed to short the Instream Flows that are senior to or at the priority implied by the controlling date.
- Again execute rules to operate reservoir Storage Accounts to meet other objectives.
- Reconcile the accounting solution (reservoir accounting releases, appropriations, and other accounting diversions) with the physical simulation objects (reservoir releases, diversions from reaches) and simulate the physical system

In this approach, water rights and the accounting solution drive the simulation system. This is different from the typical accounting model in which the accounting system solves after the physical system.

Creating a Model

This section describes the steps necessary to create a model that uses the water rights solver. Each type of right is described along with the methods available on both the containing object and the account. Next, the computational subbasin and its methods are described. Finally, the allocatable flow supply chain from which the solver allocates and other assumptions made about the model are discussed. In general, only an overview of the methods is presented in this document; details are provided in the reference material. Links are provided to take you to the appropriate sections of the reference material.

Storage Rights

Storage rights are modeled with Storage Accounts on any of the four reservoir objects: Storage Reservoirs, Level Power Reservoirs, Sloped Power Reservoir, and Pumped Storage Reservoirs.

Object Methods

If the capacity of the Storage Account is limited to the volume of the conservation pool, select either Conservation Pool or Conservation and Flood Pools in the Operating Levels category on the reservoir. If the None method remains selected, this physical constraint will not be enforced and the Fill Conservation Pool and Fill Conservation Pool with Diversions methods on the Storage Account will not be available to calculate initial requests. The specified operating levels are not required to match entries in the reservoir's Elevation Volume Table, however, the Elevation Volume Table must cover the range of values in the Operating Level Table. For more information on these methods, see "Bank Storage" in Objects.pdf.

Account Methods

On each Storage Accounts that has a water right, it is necessary to specify the following:

In the Water Right category, the Priority Right must be selected. This is the same as clicking on the Has Priority Date toggle on the General tab of the Open Account (configuration) dialog. Specify a unique priority date.

In the Initial Request category, select the Specify Initial Request, the Fill Conservation Pool, or the Fill Conservation Pool with Diversions method to specify how the initial water right is determined. In the Specify Initial Request method, the user can specify (via input, dmi, or rules) the Initial Request slot. If the Fill Conservation Pool method is selected, initial allocations to Storage Accounts are limited to that amount which will fill up the conservation pool on the reservoir. If the Fill Conservation Pool with Diversions method is selected, initial allocations to Storage Accounts are limited to that amount which will fill up the conservation pool on the reservoir and meet diversions totalling the sum of the Initial Requests of all direct diverters from the storage account. The total volume in all Storage Accounts on a reservoir contributes to the conservation pool volume, regardless of the ownership of the water in these accounts. For this reason, in order to prevent a single storage account from holding all the water in the conservation pool, the Conservation Pool Fill Factor category may be used. This category contains methods that allow a fraction of the conservation pool (and diversions) to be allocated, thereby reserving some for other accounts. See "Initial Request" on page 18 for more information.

If lagged return flows entering the Storage Account are to be credited to the appropriation from allocatable flow, select the method Return Flow Credit in the Appropriation Request Adjustment category.

Storage rights might be subject to legal constraints as well as to physical constraints. Examples of legal constraints are minimum bypass constraints and accrual-based maxima. If there is a minimum bypass, select a method in the

Min Bypass category. Currently available is the Fraction of Flow Above Min method. See “Min Bypass” on page 24 for more information on this method.

When the Fill Conservation Pool with Diversions method or the Fill Conservation Pool method is selected, the solver saves a copy of the resulting Appropriation Request slot value in the Initial Request slot, if the solver finds that the Initial Request slot value is not already defined. This use of the Initial Request slot differs from the standard usage in that its value reflects the application of legal and physical constraints, and in that it is not computed on every call to the solver.

Linking Accounts

Allocations to Storage Accounts on in-stream reservoirs from the allocatable flow supply chain on the same reservoir are made through “transfer” type of links (supplies) as shown.

Figure 5-2:

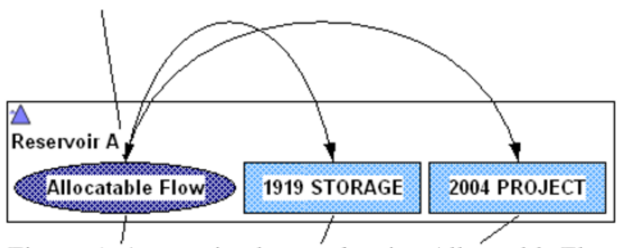


Figure: 1. Accounting layout showing Allocatable Flow supply chain being transferred to a storage right on the same reservoir

Off-stream reservoirs with storage rights receive allocations diverted from the main river or an in-line reservoir. They must be linked to the Reach or Reservoir Object through a Diversion Object. The allocation is subject to the physical capacity of the diversion structure at the point of diversion (Diversion Capacity slot on the Reach). Below is a view of this setup. Water is allocated from the Allocatable Flow supply chain to the 1920 Storage right, but it must go through a passthrough account on the DIVOBJ Diversion Object. The passthrough on the Diversion Object is linked to the Allocatable Flow passthrough account on the Reach via a “Diversion” type of supply (link). The Inflow of the Storage Account on the offstream reservoir gets a supply from the Outflow of the passthrough account on the Diversion Object.

Figure 5-3:

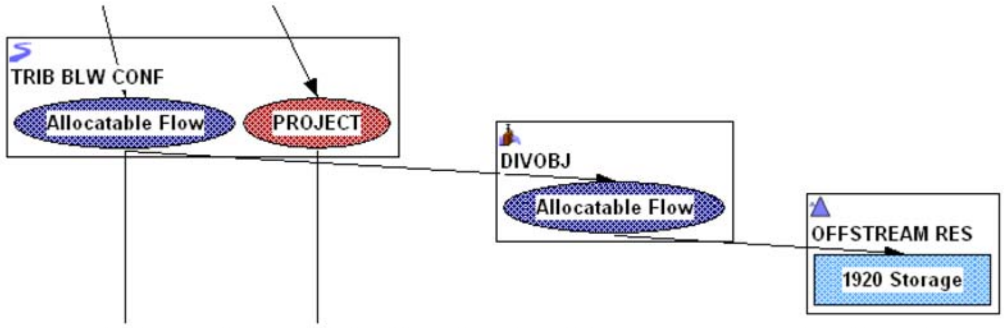


Figure: 1. Accounting layout showing Allocatable Flow supply chain being diverted to a diversion object and then to the off-stream reservoir

You can have multiple water rights on the offstream reservoir. There should be one passthrough account on the diversion object **for each** water right on the reservoir. The water type of the passthrough account on the diversion object does not matter. It can be the same or different than the allocatable flow chain.

Warning: Storage accounts should not be linked into the allocatable flow supply chain by Inflow/Outflow supplies. In allocating to a Storage Account, the solver will not include the volume of diversions or releases from the Storage Accounts at the current timestep. Such policy decisions are expected to be addressed with rules.

Diversion Rights

Diversion rights are modeled by a Diversion Account on a Water User or AggDiversion Site Object.

WaterUser and Aggregate Diversion Site Object Methods

No additional method selections on the object are required to use the solver.

Diversion Account Methods:

In the Water Right category, the Priority Right must be selected. This is the same as clicking on the Has Priority Date toggle on the General tab of the account configuration dialog. Specify a unique priority date. In the Initial Request category, select one of the Specify Initial Request, Disaggregate by Subbasin, or Max Permitted methods to specify how the initial water right request is determined. Click “Initial Request” for more information.

For water rights allocation, diversion accounts may only use the Simple Lag method in the Return Flow Route or Split “Return Flow Route or Split”. The Split and Route method cannot be used.

Diversion rights might be subject to legal constraints as well as to physical constraints. Legal constraints include:

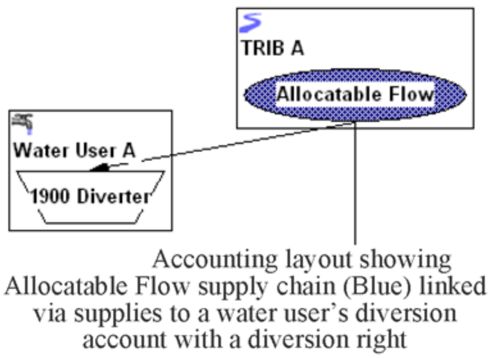
- Max Legal Request: This method allows the user to enter a value for the Initial Request that is larger than the legal right. This is useful if you are computing the Initial Request from aggregated or historical data. For more information, click “Max Legal Request”.

- Min Bypass: If there is a minimum bypass, select a method in the Min Bypass category. Currently available is the method Fraction of Flow Above Min. Click “Min Bypass” for more information on this method.

Linking

Allocations to a Diversion Accounts is limited by the value in the Reach’s ***Diversion Capacity*** scalar slot. If no value is given, no physical constraint is applied. As shown, the Diversion Account’s Diversion slot gets a supply from the Diversion slot on the passthrough account on the Reach.

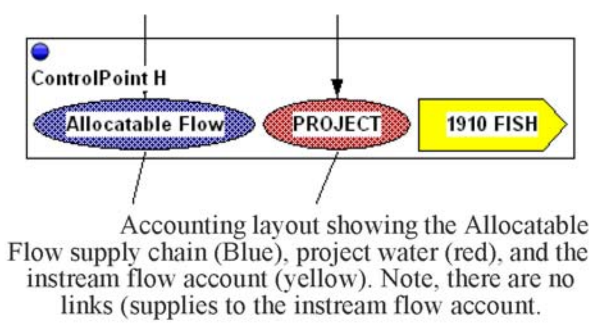
Figure 5-4:



Instream Flow Rights

An instream flow right is modeled by an Instream Flow account on a Control Point Object. Shown is a a 1910 FISH Instream Flow Account. No supplies are necessary to link this account to other accounts on the Control Point; the Flow slot of the Instream Flow account contains the sum of the Inflows to all the passthrough accounts on the object.

Figure 5-5:



Control Point Object Methods

One method category on the Control Point Object contributes to the computation of Initial Request based on storage in upstream reservoirs. This method is on the object rather than the account because it uses information from other simulation objects. The category is Instream Flow Reference Level, from which the user can select the Reservoir Storages Lookup method. This method allows the user to establish a reference level on which the Instream Flow Account can base its water allocation requests. This method executes one timestep each year identified by the Start of Reference Year slot. At that timestep, this method sums the storages at the previous timestep from each of the Reference Reservoirs. The sum is then looked up in the Reference Level Storage Table to determine the Reference Level. The Reference Level is used by the Instream Flow accounts on the object to determine the account's initial request. For more information, see "Reservoir Storages Lookup" in Objects.pdf. (Objects/ Control Point/ User Methods/ Instream Flow Reference Level)

Instream Flow Account Methods

In the Water Right category, the Priority Right must be selected. This is the same as clicking on the Has Priority Date toggle on the General tab of the account configuration dialog. Specify a unique priority date

In the Initial Request category, select either the Specify Initial Request or the Based on Reference Level method to specify how the initial water right is determined. See "Initial Request" on page 47 for more information.

The behavior of the Instream Flow Accounts during the water rights solution depends on the controlling date argument for the SolveWaterRights() or SolveWaterRightsWithLags() function call. The controlling date is one that determines which Instream Flow Accounts can make calls and which only compute their Available Allocatable Flow slot values. The accounts whose priority dates are equal to or earlier (i.e., are senior to) the controlling date are allowed to place calls on junior upstream rights. The call is limited by the value in the already-computed Available Allocatable Flow slot of the Instream Flow Account. Thus, the first call to the solver must have had a controlling date senior to all Instream Flow Account priority dates.

Computational Subbasin

A computational subbasin is defined by the user to specify the objects which are to be part of the water rights solution. On the computational subbasin, select the Prior Appropriation method on the Water Rights Allocation category. (There is only one solver, however, the category allows for future implementation of other allocation schemes.) For a description of the slots associated with this method, see "Water Rights Allocation" in Objects.pdf. Once this method is selected, a new category, Account Initial Request, becomes visible. This category allows the user to specify a method to disaggregate annual requests. The available methods are **No Method**, Periodic Coefficients or Series Coefficients; the user should select the method based on the timestep of the model and the preferred way to input the coefficients. For more information on these methods, see "Account Initial Request" in Objects.pdf.

Data Requirements for the Allocatable Flow Supply Chain

The following model assumptions must be met for the solver to work correctly:

- There is no way for water to leave the allocatable flow supply chain from which this the water rights solver allocates other than through:
  - The allocations made by the solver
  - Gain Loss slot values which are output. This is accomplished by selecting the appropriate Gain Loss Coefficient method (See "Gain Loss Coefficient" on page 55) on passthrough accounts in the chain and then specifying the coefficient (either scalar, series or periodic) if desired. If Gain Loss slot values are not determined by a gain loss coefficient (i.e. they are user input), the solver cannot solve upstream for cutbacks to upstream appropriations.

- In addition, inflows, outflows, non-appropriation transfers, and return flows are assumed to be non-negative throughout the supply chain and there are no existing diversions or (appropriation) transfers to accounts that will not be re-allocated by this method. Slot inflow values may be negative only to the extent that they do not produce negative outflows anywhere in the supply chain at any time during the solution. See the note below for the only exception to this.

- At Instream Flow Accounts, no sibling accounts have negative inflows. If they do, this method cannot solve upstream for cutbacks to upstream appropriations. See the note below for the only exception to this.

Note: Negative flows at the time of allocation are allowed only if the Allow Negative Flows method is selected on passthrough accounts for the allocatable flow chain on reaches, reservoirs and control points.

- Because the solver relies on the account-solving infrastructure, certain required input values must be known on the accounts so that the supply chain can solve before this method is called. The required knows for account solution are documented in the Accounting reference documentation. (See “Accounts” on page 13.) The following is a summary of the slot requirements for this solver to work:
  - On Passthrough and Instream Flow Accounts: Inflow slot values will be propagated from above, except for those at the heads of the tributaries of the allocatable flow supply chain, which do not require Inflow slot values.
  - Passthrough accounts not on a reservoir: Slot Inflow must be defined. Gain Loss slot values may not be user input (they may be Output). But a gain loss coefficient (either scalar, series, or periodic value based on selected method) may be defined in which case, if Gain Loss is NaN, the passthrough accounts will compute the Gain Loss slot values.
  - Passthrough accounts on a reservoir with storage allowed: Inflow and Slot Inflow must be defined.
  - Passthrough accounts on a reservoir with storage not allowed: Slot Inflow must be defined.
  - Diversion accounts: Nothing required a priori.
  - Storage accounts: Slot Inflow and Gain Loss must be defined.
  - Instream flow accounts: Nothing else required.
  - If passthrough accounts at the tops of the tributaries’ allocatable flow supply chain can solve (by virtue of having Slot Inflows, these accounts will produce the flow slot values required for the rest of the supply chain to solve before this method is called.

Common Modeling Errors

This list describes some common modeling errors we have seen during the building of test models for testing this method. We have included them for your convenience.

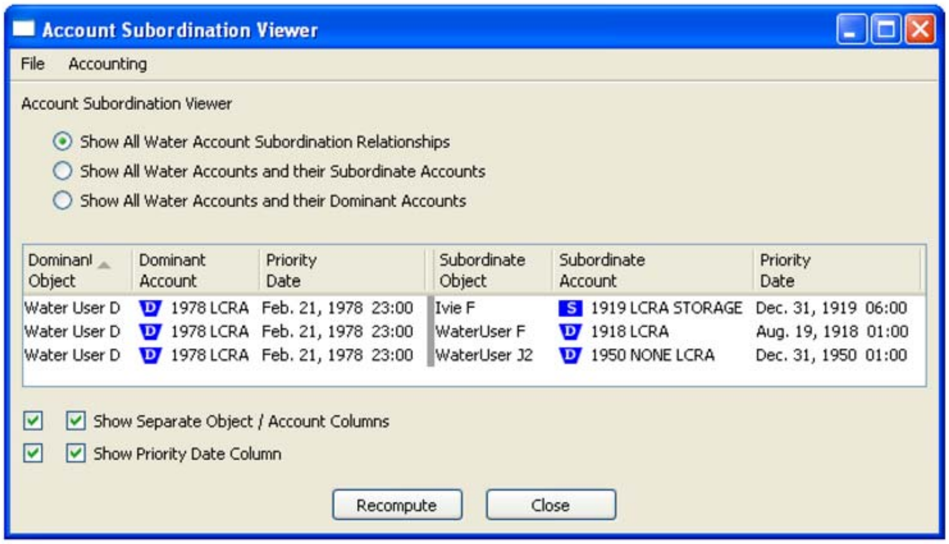
- Adding an object to the model and forgetting to put it into the subbasin.
- The subbasin is disabled. Use Workspace->Open Computational Subbasin followed by Subbasin->Enable Subbasin.
- Model uses the wrong controller. Set your controller to Inline Rulebased Simulation and Accounting.

- Unintended convergence criteria on account slots. Check your convergence criteria.
- Wrong units on account slots. Make sure all the units are what you expect.
- Lags on reaches do not match lags on their accounts.
- Forgetting to set the water type on the supplying account of a water right (at the point of appropriation from the stream). The water type must be set at each point of appropriation from the stream. If you see an Appropriation Request slot value being calculated for a right, but no allocation made to that right (not even an allocation of zero), this is likely to be the reason. Use the visualization tools on the accounting workspace (i.e. Account Groups) to display accounts with a given water type, as in Figure 1, above. Then change any incorrect accounts to the right water type.
- Having more than one account on an object with the allocatable flow supply type. The results are undefined when this is the case. There is one exception, for offstream diversions (described in “Linking Accounts” on page 175), the Diversion object can have multiple passthrough accounts with the allocatable flow supply type.
- User-input data for allocation requests was put into the wrong slot: it should go into Initial Request; not Appropriation Request. The Appropriation Request slot is for outputs of the solution method; this is where the method writes the request as constrained by the state of the system “at priority time”, i.e., after senior accounts have been satisfied.
- Attempting to use Fill Conservation Pool or Fill Conservation Pool with Diversions method on a Storage Account without setting up the necessary methods and data on the underlying Reservoir object.
- Too many hops from the allocatable flow (stream) to the water right account.
- Supplying a reservoir with “project” water before the solver is called, thus reducing the appropriation to storage.
- Failing to clear supply chains from non-allocatable water before calling the solver. These allocations may reduce allocations to storage from allocatable water, and they may reduce diversions where diversion capacity is limiting.
- Not providing enough information for all accounts in the subbasin to solve once before the solver rule executes. See the required data in the previous section. This data can be specified by Inputs, Rules, or Object Level Accounting methods. (See “Object Level Accounting Methods” on page 66.) In most models, you probably want to use the
  - Zero Slot Inflows method (see “Zero Slot Inflows” on page 69) for most objects. This should be configured to execute at “Beg of the Run”.
  - Copy Slot to Slot Inflows method (see “Copy Slot to Slot Inflows” on page 69) for local inflow reaches, control points and reservoirs. This should be configured to execute at “Beg of the Run” or beginning of timestep depending on when the local inflows are known.

Subordination Viewer

After setting up many subordination relationships between accounts, it can be difficult to keep track of the relationships. The Account Subordination Viewer assists the user in viewing these relationships. This dialog is accessed from the Water Accounts Manager using the System > Subordinated Rights... menu. Shown to the right is an example of this dialog. In the list view, the left side shows the dominant accounts, the right side shows the subordinate accounts.

Figure 5-6:



The dialog has three options for displaying the relationships as defined by the radio buttons:

- Show All Water Account Subordination Relationships: Show ONLY those accounts involved in a subordination relationship
- Show All Water Accounts and their Subordinate Accounts: Show all accounts on the left side and their subordinate accounts on the right. Accounts may be repeated.
- Show All Water Accounts and their Dominant Accounts: Show all accounts on the right side and their dominant accounts on the left side. Accounts may be repeated.

At the bottom of the dialog, there are check box options to Show Separate Object / Account Columns for both the left and right sides. The user can either show a column for the Object and another for the Account or one column that lists Object^Account. This option is useful for sorting through accounts or objects. The user can sort the list by any of the columns in this dialog by clicking on the column heading.

There is also an option to Show Priority Date Column for both the left and right sides. This allows the user to hide or show the Priority Date column.

Finally, this dialog is strictly a viewer, but using the right-click context menus, the user can Edit or Configure the account or open the Object containing the account. If the subordination relationship changes (outside of this dialog), the user can click the Recompute button to refresh the display.

Solution Algorithm for SolveWaterRights()

This section provides a detailed description of the solution algorithm for the function SolveWaterRights() and SolveWaterRightsWithLags(). SolveWaterRights() can only be called on a subbasin with no lags on the passthrough accounts in the subbasin, and so when it is used, all ***Local Timestep Offsets*** are zero, and the solution involves data at the current rules-controller timestep only. When SolveWaterRightsWithLags() is called, Local Timestep Offsets may be greater than zero, and the solution involves data at timesteps which may differ from the current rules-controller timestep.

The rule function's subbasin argument identifies a computational subbasin with a method selected in the Water Rights Allocation category. The Prior Appropriation method is described here. The method does the following:

1. Determine local timestep of the accounts representing the rights. (See "Determine the Local Timesteps of the Rights" on page 182.)
2. Clone the accounting network. The solver works on this cloned system to solve the problem. (See "Cloning" on page 185.)
3. Clear values on supplies that represent allocations from the allocatable flow supply chain. (See "Clear values" on page 185.)

For each water right in priority order:

4. Compute the appropriation request (See "Computing Appropriation Requests" on page 186.)
5. Compute allocation that meets the request, constrained by not violating senior rights.(See "Computing Appropriation" on page 189.)

Then,

6. Create a list of {slot name, value} pairs or a list of {slot name, date-time, value} triplets of allocations that are returned by rule function. (See "Return {Slot Name,Value} or {Slot Name, DateTime, Value} List to Rule" on page 198.)

The following sections describe these steps in detail.

Determine the Local Timesteps of the Rights

The local timestep of an account representing a right is defined to be the timestep of the passthrough account representing the appropriation point of the right. For this purpose, the appropriation point of an instream flow right is the sibling passthrough account on the supply chain for which the solver is called. Because the supply chain of interest is not known until run-time (when the predefined rules function SolveWaterRightsWithLags() is called), the local timesteps of the accounts representing the rights must be computed by the solver.

When SolveWaterRights() or SolveWaterRightsWithLags() is called on a subbasin that contains no lags, all the Local Timestep Offsets are zero, and the local timesteps of the rights are the current rules control timestep. Note that the entire subbasin must contain no lagged passthrough accounts for this to be true.

The local timestep of the appropriation point is a timestep in the future, defined as current rules-controller timestep + Local Timestep Offset timesteps. All solver operations on the account, its sibling accounts, its appropriation point account and the appropriation point's siblings, and the underlying simulation object will take place at this local timestep. The Local Timestep Offset slot values of the passthrough accounts have been computed at the beginning of the run. The process of determining these offsets is described below.

At the Beginning of a Run

At the beginning of a run, before the first timestep, an enabled computational subbasin with the Prior Appropriation method selected will compute the Local Timestep Offset slot values for necessary passthrough accounts on objects in the subbasin.

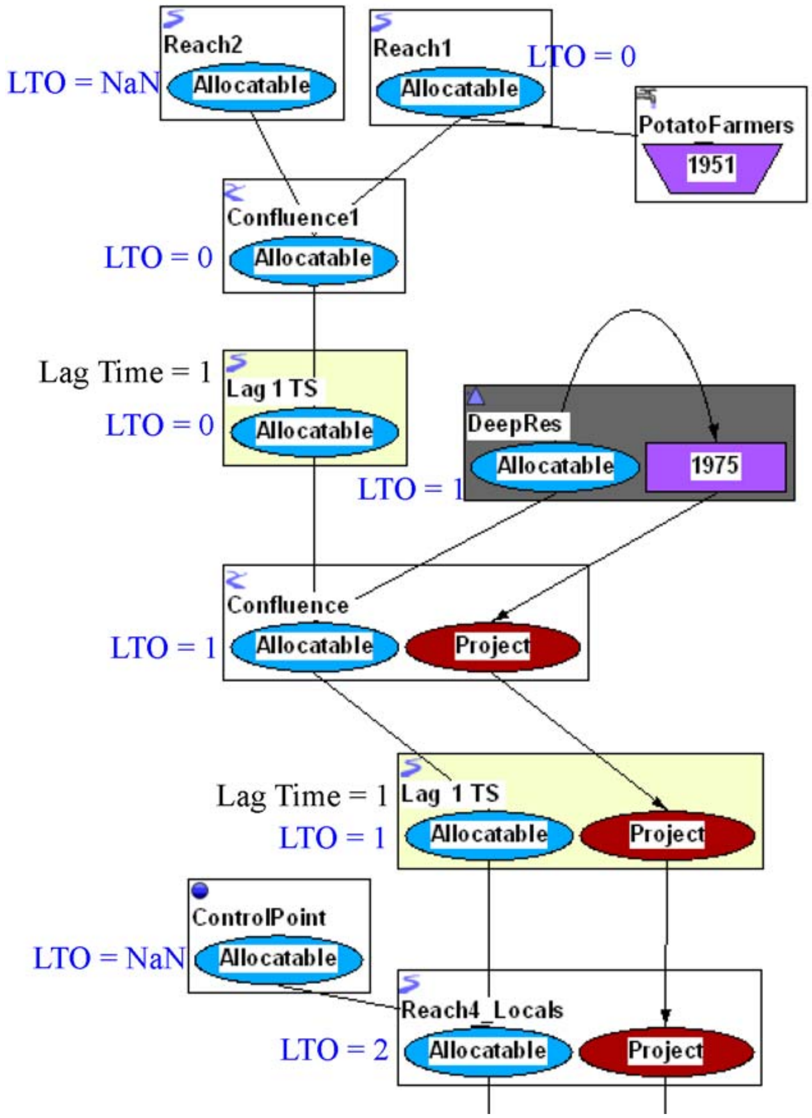
The computation is not system-wide; it's a computation done on the water-rights subbasin, and it's restricted to the part of the subbasin containing rights that appropriate within the subbasin. Thus, only the accounts that figure in the solution for the subbasin are given the offsets. If there's no appropriation at or upstream of a passthrough account on its chain, then the passthrough doesn't need a local timestep and is not reset. If you desire to use the Local Timestep Offset in rules or OLAMs and it is not set by the solver, you should set it yourself.

The computation of the Local Timestep Offset takes place in steps:

1. Compute maximum lag to each passthrough account. This is the largest cumulative lag time (in number-of-timesteps) from any headwater account to the passthrough account, considering only passthrough account lag times (not return flow lags).
2. Compute Lag Distance of the subbasin. The lag distance of the subbasin is the largest maximum lag of all the passthrough accounts in the subbasin; it represents the longest travel time through any supply chain in the subbasin.
3. Compute Local Timestep Offset of the passthrough accounts. The local timestep of an account is the lag distance of the subbasin minus the lag, in timesteps, from the account’s inflow to the end of the subbasin. If lags on supply chains do not match, this can give counter-intuitive Local Timestep Offsets, which can be corrected by adding “dummy” lagged passthrough accounts at the ends of the supply chains with lower net-lags.

To the right is an example subbasin that shows the Local Timestep Offset computation. In this example, there is a 1 timestep lag at the two yellow reaches (Lag 1TS). Thus, the subbasin’s Lag Distance is 2. The resulting Local Timestep Offset (LTO) are shown. Note, that the LTO at Reach1 and ControlPoint are NaN as they are not needed by the solver.

Figure 5-7:



In the Solver Rule Function

During a call to the solver, the water rights use a local timestep that is determined as follows:

- A Diversion Account uses the local timestep of the passthrough account that represents the (single) appropriation point from the supply chain for which the solver is called. This is an account from which a diversion is made into the diversion account, and has the water type given as an argument to the solver.
- A Storage Account uses the local timestep of the passthrough account representing the (single) appropriation point from the supply chain for which the solver is called. For an in-stream storage right, this is an account from which transfers are made into the storage account, and it has the water type given as an argument to the solver. For an off-stream storage right, this is a passthrough account from which diversions are made into a

passthrough account on a Diversion Object, and through which the paper water flows into the off-stream storage account's Inflow slot.

- An Instream Flow Account uses the local timestep of the sibling passthrough account that has the water type indicating the supply chain for which the solver is called.

Note: *Note: Some Solutions Involve Other Timesteps*

A Passthrough Account may have data requirements for timesteps other than its local timestep. Two examples of such cases are when objects have more than one supply chain and the lags on the supply chains differ. In such a model, solving for the Flow slot on an Instream Flow Account requires summation of the inflows across all sibling accounts at the local timestep of the Instream Flow Account for a given Water Type, which may differ from the local timestep of one or more of its sibling accounts. Similarly, solving for the Appropriation Request on a storage account requires summation of the inflows (including transfers in) into all the sibling storage accounts at the local timestep of the storage account whose Appropriation Request is being solved.

Note also that inasmuch as account methods use data from the underlying simulation object at local ("future") timesteps for the accounts, the simulation objects must have dispatched at these future timesteps before the solver is called. An example is the storage account's Fill Conservation Pool method for computing the Appropriation Request. This method uses the underlying reservoir's Conservation Pool Initial Empty Space slot value at the storage account's local timestep. If the local timestep offset of a storage account's appropriation point is non-zero, the value of the Conservation Pool Initial Empty Space will be needed before it is computed by the beginning-of-timestep phase of the run controller, so the solver will force the computation of the value if needed. The beginning-of-timestep method for the reservoir will not recompute the value once it is defined. The reservoir must have dispatched at the prior timestep to provide a value for Storage, which is needed to compute the Conservation Pool Initial Empty Space.

Cloning

When the rule function is called the first time in a run, it clones all the accounts in the subbasin and their supplies (links). Using the cloned network, the solver can utilize the standard account solution methods without the risk of having side effects on the "real" accounts, and it can store intermediate results. Each time the solver is invoked by the rule function, current values from the "real" account slots are copied to the slots on the clones for the set of timesteps necessary to compute the solution. Thus, the rule function abides by the dictate that rules have an "all-or-nothing" effect and rule functions do not change the state of objects.

As the clones are constructed, tables are created to map donor accounts to their clones. The computational subbasin maintains this table. The cloned accounts reside on the computational subbasin because their underlying objects are not cloned, in the interest of performance. When the solver has to determine which cloned accounts are siblings (their donors reside on the same simulation object), it refers to the mapping.

Clear values

Each time the rule function is called, in the cloned network it clears all supplies that represent allocations to water rights accounts. Only these supplies are cleared, and not other supplies. With reference to Figure5-7 above, a call to the solver using the water type "Allocatable Flow" will clear (in the corresponding cloned accounting network) the following supplies:

- Bartlett Reservoir ^ Allocatable Flow to Bartlett Reservoir ^ 1922 Storage
- Bartlett Reservoir ^ Allocatable Flow to Bartlett Reservoir ^ Farmers Project
- Bartlett to Border ^ Allocatable Flow to Green Valley Diversions ^ 1902 Diverters.
- Muddy Reservoir^ Allocatable Flow to Muddy Reservoir ^ Project.

Computing Appropriation Requests

The solver “visits” each account in priority order: most senior first, and computes appropriation request, then the appropriation. This section describes the computation of the Appropriation Request.

The appropriation request is the amount that the account is legally entitled to ask for. It is the amount that the solver tries to allocate, subject to availability of water not already taken by senior rights. The appropriation request is derived from an initial request, which might be known a priori and taken from the Initial Request slot, or which might have to be computed, based on the state of the network. The account’s method selection in the Initial Request category determines how the initial request is obtained. If the Initial Request slot becomes visible as a result of the method selection and the initial request is computed, its value is stored in the slot. (Whenever the Initial Request slot is visible, so is the Shortage slot.)

Legal and physical constraints may apply to the initial request value, producing a reduced value for the appropriation request, which is stored in the Appropriation Request slot. The legal and physical constraints must reflect the state of the network at “in-priority time,” namely, after all seniors have received their appropriations.

Computing Appropriation Request from Initial Request

The solver does this in steps:

1. Compute physical constraints applicable to the allocation, based on the state of the cloned network at this time. The solver does this before having the account do anything, because the solver “knows” about the physical network, which the account does not. The solver determines an upper bound on the allocation due to the physical constraints. Specifically,
    - *Reach Diversion Capacity:* If the water right account is a diverter from a reach, the capacity of the diversion structure on the reach limits the diversion. All other diverters from the same reach share this structure, so the upper bound on the Appropriation Request is the Reach object’s Diversion Capacity minus the sum of all existing diversions from the reach’s passthrough accounts. (Remember that before we got here in the solver, this appropriation was cleared, so its diversion is zero.). The reach’s Diversion Capacity slot is used to determine the diversion capacity, and if its value is not defined, no diversion capacity limit is applied.
    - *Reservoir Conservation Pool:* If the account is a storage right, the remaining space in the reservoir’s conservation pool limits the Appropriation Request. The inflow required to fill the conservation pool is taken from the underlying simulation object: The reservoir’s slot Conservation Pool Initial Empty Space is computed at the beginning of each timestep as the Inflow volume that would be required to bring the reservoir to the top of conservation pool. It is computed by executing a method on the reservoir “solve inflow given storage and outflow.” The method assumes that outflow is zero and includes evaporation in the mass balance. This flow is used by the solver to determine how much flow it can add, through appropriation, to the total flows into Storage Accounts on the reservoir. All of the reservoirs’ Storage Accounts’ Inflow, Slot Inflow and Transfers In slots are summed; all Transfers Out are subtracted, and

the resulting flow is the amount of flow already going into the Storage Accounts. This value is subtracted from the conservation pool initial empty space, to determine an upper bound on the appropriation.

- Both of these physical constraints might apply, so the lower of the two upper bounds is chosen.

2. Execute Appropriation Request Methods on Water Right Account: The solver asks the account to compute its appropriation request, applying the upper bounds on the request due to physical constraints: Call the C++ utility method computeDemandForWRA on the account. This method has three steps:

   a. First invoke the account's user-selectable method for determining its ***initial request*** as described above, and store this value in a temporary place, Then the upper bound is applied, possibly reducing the request, and the result is again stored in a temporary place.

   b. Second, invoke the account's user-selectable method for ***Appropriation Request Adjustment,*** if any (available method is the ***Return Flow Credit*** method on Storage Accounts). Pass in the temporary result from above, and store the reduced request again in a temporary place.

   c. Invoke the account's user-selectable methods for legal constraints, including `Max Legal Request` and `Min Bypass`.

      - `Max Legal Request` allows the user to enter a value for the Initial Request (on Diversion Accounts only) that is larger than the legal right. This is useful if you are computing the Initial Request from aggregated or historical data. If there is a valid `Maximum Request` (not NaN), then:

$$\text{reduced request}[date] = Min(\text{Initial Request}[date], \text{Maximum Request}[date])$$

Note: *The reduced request is constrained to be non-negative. For more information, see* *"Max Legal Request" on page 43..*

- `Min Bypass` constraints require storage and diversion rights to leave a prescribed flow in the stream at one and only one reference (measurement point), or a percentage of the flow at that reference point, or a combination of absolute flow + percentage. The default reference location is the point of appropriation for the right (a passthrough account on a Reach or Reservoir). An alternate location can be specified, but it must be a Control Point. It can be given in the Bypass Reference Location, which is a list slot that can refer to an object.

The input data needed to compute the MinBypass are:

AbsMinBypass = Absolute minimum required (or zero) to be left in the river at the ref point

FractionAboveMin = Fraction of flow required to be river (or zero), or fraction above the AbsMinBypass if that value is greater than zero.

The Minimum ByPass, the amount required to be left in the stream at the ref point is calculated as:

```
MinBypass = Min((AbsMinBypass+(TotalFlow-AbsMinBypass)*FractionAbove-
Min),
        TotalFlow)
```

Where the total flow is the flow at the reference point before the allocation is made.

At the default reference location (Reach or Reservoir), the Total flow is

```
Total Flow = sum of Available for Appropriation on all passthrough accounts
```

If the reference point is an alternate location, i.e., a Control Point, the Total Flow is

```
Total Flow = sum of Outflow of all accounts
```

The Upper Bound on the Appropriation Request due to Minimum ByPass is

```
UpperBound(at point of reference) = Total Flow - Minimum Bypass
```

This is the Upper Bound that the solver applies if the reference location is the default, i.e., the point of appropriation. However, if the reference point is a downstream Control Point, the effects of gain/loss and lag between the reference location and the appropriation point must be taken into account when determining the resulting upper bound on the appropriator's appropriation request. To do this, the account method makes a callback to the solver, which then pushes onto a stack all the passthrough accounts on the allocatable flow supply chain from the appropriation point to the measurement location. Then it solves upstream by popping accounts from the stack and for each, it routes the upper bound through the account as follows:

UpperBound = UpperBound(from d.s.Account) / (1-GainLossCoeff)

The new upper bound is passed in as input to the solution for the next account popped from the stack. When the stack is empty, the resulting upper bound is returned to the selected method on the appropriating account, which then stores the upper bound in the Temp Min Bypass Constraint (temporary[1]) slot. Hence, another upper bound has been computed by computeDemandForWRA.

Note: *The minimum bypass constraint applies only to the account with the method selected; it does not in any way affect rights that are junior to this appropriator. If the minimum flow is required to be left in the stream by juniors, all juniors must also have the same method selection and slot values to model the constraint.*

---

3. After these 3 possible constraints on the Appropriation Request have been computed, the solver *computes the final Appropriation Request* by applying the constraints to the Initial Request and writing the resulting request value to the Appropriation Request slot.
4. If the water right account is an Instream Flow Account on a Control Point object, and it is junior to the controlling date on the solver call, then computeDemandForWRA copies the current Outflow slot value on the allocatable flow supply chain passthrough account on this Control Point into the Available Allocatable Flow slot on the Instream Flow Account.

1. This temporary slot is not stored in the model file, and is present for debugging purposes only, and may be removed from RiverWare at any time: do not write rules that use it.

Computing Appropriation

There are two ways to compute the appropriation, with and without subordination. Depending on method selection, without subordination may also include special calculations for accounts sharing a priority date. Computing appropriations without subordination is described first, followed by the modifications used to model subordination.

Computing Appropriation without Subordination

To compute the appropriation for a storage right or diversion right, the solver first proposes an appropriation, then determines if this appropriation would short downstream seniors, If so, the appropriation must be cut back, and the solver determines how much it must reduce the appropriation. The cutback results in the maximum appropriation that is legally permitted. The following describes the details of the solver's logic in this process. The next section describes the logic of the solver when subordination relationships exist between rights in the model.

Note: *When negative flows are allowed: Before proposing an allocation, each passthrough account with the Allow Negative Flow method is visited and the Temp Available Before Appropriation is set to the value of the Temp Available For Appropriation. This is done to record the condition of the system before each right is visited. If the available is already negative, then any allocation to upstream juniors cannot make it more negative, but will not try to undo the negatives.*

The solver proposes an allocation to the Diversion or Storage Account by setting its supply from the designated allocatable flow supply chain passthrough account at the appropriation point to be the smaller of:

- the Storage or Diversion Account's Appropriation Request slot value,

and

- the water available in the allocatable flow passthrough account as given by the passthrough account's temporary slot Temp Available For Appropriation. This temporary slot reflects appropriations from the allocatable flow supply chain to senior rights by the current invocation of the solver. The Available for Appropriation is computed each time the passthrough account solves in the cloned network created by the solver. It reflects the available water in the passthrough account including the sum of inflows, slot inflows, transfers in, transfers out and diversions on that account; it does not include the account's gain/loss, lag, or return flows.

Note: *The Shortage slot on the Storage or Diversion Account does not bound the appropriation. Thus supplies other than allocatable flow (e.g. releases from upstream storage) that help to satisfy the request do not reduce the allocatable flow appropriations.]*

As soon as the solver sets the supply that diverts the proposed allocation from the allocatable flow supply chain, the cloned accounting network solves. The diversion of water from the allocatable flow supply chain could possibly cause negative flows in downstream passthrough accounts where senior rights have already diverted water. These negatives are reflected in the Temp Available For Appropriation slot on downstream passthrough accounts in the allocatable flow supply chain. A negative flow indicates that the proposed appropriation would "short" a downstream senior storage or diversion right. The magnitude of the shortage is:

- When negative flows are not allowed, the shortage is the absolute value of the negative flow found in the Temp Available For Appropriation slot.
- When the Allow Negative Flows method is selected on the passthrough account, the shortage is computed as: Abs(Temp Available For Appropriation - min(Temp Available Before Appropriation,0)). This modification is made so that if the available before appropriation is already negative, then the shortage is computed as the additional impact by this upstream allocation, not the total negative flow.

For senior downstream instream flow rights, a shortage manifests as a positive value in the Instream Flow Account's Shortage slot.

The solver inspects the shortage at allocation points for Diversion and Storage Accounts, and checks the ***Shortage*** slots of Instream Flow Accounts only for the rights that are senior to the account for which the proposed appropriation was just made. The solver does not check for shortages at points between senior downstream appropriation points, nor does it check for shortages at the bottom of the subbasin, if the bottom lies downstream of the most downstream senior right. (The solver has tables to direct it to the most-downstream senior in an expedient manner.)

If, upon inspection, the solver finds that there is no induced shortage for any downstream senior right, the proposed appropriation stands, and the solver moves on to visit the next-in-priority water right.

Note: *Based on an account's method selections, there are cases in which accounts record the amounts they could have appropriated, but do not make the appropriation. For these accounts, the resulting appropriation is recorded in the **Appropriation Request** slot and the proposed appropriation is set to zero before the solver moves to the next-in-priority account.*

If, however, the solver finds that there are any downstream senior shortages induced by the proposed appropriation, the solver identifies the most downstream shorted senior. It pushes onto a stack all passthrough accounts in the supply chain between the proposed allocation and the most-downstream senior shortage. The initial value of the cutback to the proposed allocation is the absolute value of the shortage on the most downstream shorted senior. The solver then goes through accounts from downstream to upstream, propagating the shortage upstream and applying gains/losses, to determine how much the proposed appropriation must be cut back to result in no shortages to downstream seniors. (The use of the stack enables the solver to solve upstream back to the appropriation point, and take the proper branch when solving upstream in the face of confluences of any sort.)

The solver visits each passthrough account on the stack in downstream to upstream order, keeping track of the cumulative shortage (CumShort). As it pops a passthrough account from the stack, it applies gains/losses to the CumShort that is propagating upstream. If the passthrough account is an appropriation point for a senior diverter, the gain/loss of the passthrough account is applied to the CumShortage, then the solver checks for a shortage as computed above. If there is a shortage value, the solver compares its absolute value with the current value of CumShortage, and takes the greater value as the CumShort, then continues to the next upstream passthrough account. If the passthrough account is an appropriation point for an Instream Flow Right, additional checks must be made to determine the correct CumShortage value. (See following paragraph.) When the solver reaches the most upstream account on the stack, which has the current proposed appropriation, the CumShort shortage is the amount that the proposed appropriation must be cut back in order to honor the downstream senior rights.

If, as it pops the passthrough accounts off the stack, the solver encounters a passthrough account on a Control Point object that has a senior Instream Flow right[1], it checks the ***Shortage*** slot on the Instream Flow Account (IFA). If ***Shortage*** > 0, then several criteria are examined to determine the correct amount to be added to the CumShortage. The IFA's "call" is the minimum of:

- ***Shortage***; (positive value; amount by which the IFA right is not met);
- the ***Appropriation Request*** + Allocatable Flow Deficit where Allocatable Flow Deficit is the absolute value of a negative ***Outflow*** of the passthrough account;
- the ***Available Allocatable Flow*** + Allocatable Flow Deficit
  where the ***Available Allocatable Flow*** is a slot whose value was recorded as the outflow of the passthrough account when the IFA was in priority; and
- Delta NF = ***Available Allocatable Flow*** **-** current ***Outflow*** value of passthrough account.

Following is a pseudo-code version of the solver's logic for checking if downstream seniors are shorted as a result of the proposed appropriation: (PTA = passthrough account)

```
CumShort = 0.0 (maintained as a positive value, hence the use of the ABS operator)
For each passthrough account on the Stack in d.s. to u.s. order:
    If (PTA does not have a senior appropriation OR
        PTA has a senior Div appropriation AND TempAvailableForAppropriation >= 0)
    CumShort = (CumShort)/(1-GainLossCoeff)

    Else if PTA has a senior Div appropriation AND shortage < 0)
    CumShort = Max{(CumShort)/(1-GainLossCoeff), ABS(shortage)}

    Else if PTA is on an object with a senior Instream Flow Acct (IFA) appropriation
        If Shortage > 0
            Call = Min (Shortage,
                Appropriation Request + NatFlowDeficit
                Available Allocatable Flow + NatFlowDeficit,
                DeltaNaturalFlow )
            CumShort = Max{CumShort, Call)
        Endif
```

The CumShort is reset to the "call" if the "call" is larger than the current CumShort. When the stack is empty (the solver has reached the most upstream account on the stack), the CumShort is the amount that the proposed appropriation must be cut back. The algorithm then reduces the proposed appropriation by this amount, which causes the cloned accounting system to solve and moves on to the next prioritized account.

1. Instream Flow Accounts reside on Control Points, which do not support any sort of diversion, so an appropriation point cannot serve *both* an instream flow right and a diversion or storage right.

Computing Appropriation without Subordination For Equal Priority Accounts

If two or more accounts have the same priority date, a method must be selected under the Account Equal Priority Allocation category on the computational subbasin to specify how to handle appropriations in this case. The following section describes these computations under the Share Proportionally with Limits method. Note that this method cannot be used in conjunction with subordination and that instream flow accounts cannot be shared priority accounts.

The overall approach for this method is to iteratively share available water based on a calculated proportion equal to available flow divided by cumulative water right, with the proportion for an account limited by any downstream smaller proportion calculated for another account. The steps for this method are as follows:

1. Individually for each account in the group:
    - A proposed appropriation is calculated for the account in a normal fashion as discussed above, except the proposed appropriation is not limited by the Storage or Diversion Account's ***Appropriation Request*** slot value. The proposed appropriation will thus represent all of the available water at this account's diversion point minus any needed for downstream seniors.
    - The proposed appropriation is set into the account's ***Temp Available For Shared Priority*** slot (temporary[1]).
    - The proposed appropriation is set to zero and the cloned accounting system will resolve reflecting no water diverted for the account.
    - Process is repeated for the next account in the group, resulting in a computation for each account that is independent of the other accounts in the group.
2. For each account in the group:
    - Compute a cumulative right that is a total of the appropriation requests for this account and any in the group that are upstream.
    - Calculate a bypass value for downstream seniors that is the ***Temp Available For Appropriation*** slot value from this account's associated passthrough account minus this account's ***Temp Available For Shared Priority*** slot value.
3. Loop through the following until additional proposed diversions for all accounts in the priority group are zero.
    - For each account in downstream to upstream order:
        1. Get an adjusted available water value by taking the ***Temp Available For Appropriation*** slot value from this account's associated passthrough account and subtracting out the bypass value and any additional diversions made for downstream accounts in this iteration of downstream to upstream accounts.
        2. Set an available water variable to be the minimum of adjusted available water or the account's ***Temp Available For Shared Priority*** value.

1. This temporary slot is not stored in the model file, and is present for debugging purposes only, and may be removed from RiverWare at any time: do not write rules that use it.

3. Compute a proportion as the available water variable divided by the account's cumulative right.
4. Limit the proportion to a maximum of 1.0, and limit it so that it is not greater than any proportion previously calculated for a downstream account in this iteration of downstream to upstream accounts.
5. Compute an additional diversion value as the proportion times the account's ***Appropriation Request*** slot value.
6. Limit the additional diversion if it plus the proposed appropriation already set for the account will be greater that the account's ***Appropriation Request*** value.
7. Record the additional diversion value for this account for use in adjusting available water for further upstream accounts as described above.

- Set the calculated additional diversions for all accounts in the priority group onto their proposed appropriation values, which will resolve the cloned accounting system and result in new available water values for the next trip through the loop.
- Note that the loop is exited when calculated additional diversions for all accounts in the priority group are zero.

4. Check that downstream senior accounts were not shorted by the appropriations calculated for the equal priority accounts. This can happen in certain circumstances when equal priority accounts are on different branches that feed into a downstream senior. For each equal priority account:
   - Find if a downstream senior was shorted and if so, find all the equal priority accounts whose appropriation points are upstream of this senior's appropriation point and sum the appropriations for these accounts.
   - Calculate a reduction proportion as the shortage amount at the senior divided by the sum of the appropriations for the upstream equal priority accounts.
   - Reduce the appropriations for the upstream equal priority accounts by the calculated proportion. This causes the system to resolve.
   - Recheck to see if any downstream seniors of the current account are shorted. If so, repeat the above steps. If not, move on to the next equal priority account and repeat the process until all the equal priority accounts have had the downstream seniors check.

When appropriations are solved for all account's in this priority group, move on to the next priority and process as in the previous section if it is a single account or as described in this section if it is a group of accounts sharing the priority.

Computing Appropriation with Subordination

Sometimes holders of water rights make agreements that are exceptions to the strict priority-based allocation. RiverWare allows some agreements to be modelled with a subordination relationship between a senior right-holder and a junior right-holder. In such a relationship, the senior gives up allocated water to reduce a junior's shortage.

When a right is involved in a subordination relationship with another right, the algorithm for computing its appropriation is more complex than that described in the previous section. To describe the effect of subordination on the appropriation, let us define some terms ***(these terms are defined here strictly for the purpose of facilitation this discussion; they are not widely-used):***

A right S may be modeled as subordinate to a junior right J by selecting a method on the account representing J, and putting S in a list of J's subordinates. We call S the **subordinate**, and we call J the **dominant** right. J may have more than one subordinate right.

The subordination relationship is defined through method selection on each dominant account. A list is specified of the subordinate accounts on the Subordinate Rights slot. For more information, click "Subordination of Right" for the storage account and click "Subordination of Right" for the diversion account.

Appropriations for the subordinates are computed in priority order in the normal fashion (as if they were not involved in a subordination relationship), but when the appropriations for their dominant rights are computed (later, since a dominant right is necessarily junior), the subordinates' appropriations may be reduced to accommodate the dominant rights. Subordinate appropriations are reduced only if such reduction will enable the dominant right to take more, (but not more than its appropriation request). That is, the subordinate to be cut back must be upstream of the shorted right, and it must have some allocated water to give up.

To compute the best appropriation for a dominant right, the solver first proposes an appropriation, then determines if this appropriation would short downstream seniors. If so, either the dominant appropriation must be cut back or one or more subordinate appropriations must be cut back (or both). The dominant appropriation must be cut back to accommodate downstream seniors that are not subordinate, but only if no subordinate can be cut back to satisfy the non-subordinate seniors. If the most-upstream subordinate is downstream of a shorted non-subordinate senior, the dominant appropriation must be cut back. On the other hand, if a subordinate lies upstream of a shorted non-subordinate downstream senior, adjustments to the subordinate appropriation might eliminate the apparent shortage at the non-subordinate. Clearly, the downstream seniors must be considered in upstream-to-downstream order to avoid making cutbacks that have no beneficial effect.

Further complicating matters, when a dominant right has two or more subordinates, the relative priorities of the subordinates are taken into account, so that the most-junior subordinate gives up water first (if it otherwise makes sense to do so). To address this complication, the algorithm processes downstream seniors in upstream-to-downstream order, collecting **candidates** for cutbacks: these are the subordinates that have already been encountered in our upstream-to-downstream processing of downstream seniors, and that have been allocated something to give up.

Details of the Solution Logic

The following describes the details of the solver's logic.

The solver proposes an allocation to the dominant Diversion or Storage Account by setting its supply from the designated allocatable flow supply chain passthrough account at the appropriation point to be the smaller of

- the Storage or Diversion Account's ***Appropriation Request*** slot value,

and

- the water available in the allocatable flow passthrough account as given by the passthrough account's temporary slot ***Temp Available For Appropriation***. This temporary slot reflects appropriations from the allocatable flow supply chain to senior rights by the current invocation of the solver. The Available for Appropriation is computed each time the passthrough account solves in the cloned network created by the solver. It reflects the available water in the passthrough account including the sum of inflows, slot inflows, transfers in, transfers out and diversions on that account; it does not include the account's gain/loss, lag, or return flows.

Note: *The Shortage slot on the Storage or Diversion Account does not bound the appropriation. Thus supplies other than allocatable flow (e.g. releases from upstream storage) that help to satisfy the request do not reduce the allocatable flow appropriations.*

Note: *So far, this is the same as in the non-subordination case.*

Determining Resulting Downstream Shortages

As soon as the solver sets the supply that diverts the proposed allocation from the allocatable flow supply chain, the cloned accounting network solves. The diversion of water from the allocatable flow supply chain could possibly cause negative flows in downstream passthrough accounts where senior rights have already diverted water.

These negatives are reflected in the ***Temp Available For Appropriation*** slot on downstream passthrough accounts in the allocatable flow supply chain. A negative flow indicates that the proposed appropriation would "short" a downstream senior storage or diversion right. The magnitude of the shortage is:

- When negative flows are not allowed, the shortage is the absolute value of the negative flow found in the ***Temp Available For Appropriation*** slot.
- When the Allow Negative Flows method is selected on the passthrough account, the shortage is computed as: Abs(Temp Available For Appropriation - min(Temp Available Before Appropriation,0)).

For senior downstream instream flow rights, a shortage manifests as a positive value in the Instream Flow Account's ***Shortage*** slot.

The solver determines the shortage at appropriation points for Diversion and Storage Accounts, and checks the ***Shortage*** slots of Instream Flow Accounts only for the rights that are senior to the account for which the proposed appropriation was just made. The solver does not check for shortages at points between senior downstream appropriation points, nor does it check for shortages at the bottom of the subbasin, if the bottom lies downstream of the most downstream senior right. (The solver has tables to direct it to the most-downstream senior in an expedient manner.)

Note: *So far, this is the same as in the non-subordination case.*

Note: *Based on an account's method selections, there are cases in which accounts record the amounts they could have appropriated, but do not make the appropriation. RiverWare does not support this method selection on a dominant account.*

To determine which, if any, appropriations need to be cut back, the solver visits each downstream senior right, in upstream-to-downstream order. If two or more downstream seniors appropriate from the same location (passthrough account in the Allocatable Flow supply chain), these rights are visited in junior-to-senior order.

For each downstream senior visited, the solver proceeds as follows:

- Determine if the senior is a candidate for exchange
- Compute any necessary cutback to the dominant based on a shortage at the downstream senior
- Apply the cutback
- Exchange allocated water between candidates and dominant, considering only those candidates junior to the shorted senior, if the senior is a subordinate, and considering all candidates if the shorted senior is not a subordinate.

These steps are described in detail in the following subsections.

Determine if Senior is a Candidate for Exchange

If the downstream senior being visited is not one of the subordinates of the dominant, the solver proceeds as described in the following subsections.

If the downstream senior right being visited is a subordinate and is a diverter that has a non-zero appropriation, the solver puts this subordinate into a list of candidates for cutbacks, then proceeds as described in the following subsections. (If this subordinate has no appropriation, it cannot be shorted by the dominant appropriation, and it cannot exchange water with the dominant right.)

If the downstream senior right being visited is a subordinate and is an instream flow account, it is put in the list of candidates if and only if it is shorted and there are other candidates in the list. (If it is shorted, it will cause a cutback to the dominant and then, by being a candidate, it can give back all the cutback amount). If the account is put into the list of candidates, the solver proceeds as described below, but if the account is not put into the list, the solver proceeds to the next downstream senior (the Instream Flow right will suffer a shortage).

Compute Dominant Cutback

The solver now determines if the downstream senior right being visited is shorted, and if so, what is the amount the dominant right's diversion must be cut back. If the downstream senior right is not shorted, the **Dominant Cutback** amount is zero, and the solver proceeds to visit the next downstream senior right. If the downstream senior right is shorted, the solver proceeds as follows.

The solver pushes onto a stack all passthrough accounts in the supply chain between the proposed allocation and the senior shortage. The initial value of the cutback to the proposed allocation is the shortage (as described in **Determining Resulting Downstream Shortages**, above) on the downstream shorted senior. The solver then goes through accounts from downstream to upstream, propagating the shortage upstream and applying gains/ losses, to determine how much the proposed appropriation must be cut back to result in no shortages to the (non-subordinated) downstream senior. (The use of the stack enables the solver to solve upstream back to the appropriation point, and take the proper branch when solving upstream in the face of confluences of any sort.)

The solver visits each passthrough account on the stack in downstream to upstream order, keeping track of the cumulative shortage (CumShort). As it pops a passthrough account from the stack, it applies gains/losses to the CumShort that is propagating upstream.

When the solver reaches the most upstream account on the stack, which has the current proposed appropriation (the dominant account), the CumShort shortage is the **Dominant Cutback**, the amount that the proposed appropriation could be cut back in order to honor the downstream senior right. (Because we are visiting downstream seniors in upstream-to-downstream order, and making all necessary corrections, we know that there are no shortages induced at seniors above the one we are visiting, so we only need to apply gain/loss to the shortage at this downstream senior (being visited in upstream-to-downstream order) to determine the **Dominant Cutback** amount.)

Cut Back Appropriation to Dominant Right

Once the solver has the `Dominant Cutback`, it cuts back the dominant account's appropriation by this amount

```
dominant appropriation = dominant appropriation - Dominant Cutback
```

and then determines if any subordinates can be cutback better to satisfy the dominant account.

Exchange Appropriated Water with Candidate Subordinates

If the solver's candidates list is empty, there is nothing more to do with this downstream senior, and the solver proceeds to the next downstream senior account.

Note: *All of the candidates are upstream of the shorted downstream senior (except the shorted downstream senior itself) and may be senior or junior to the shorted downstream senior.]*

If the candidates list is non-empty, the candidates are inspected one at a time, in junior-to-senior order.They are cut back as much as they can be cut back to satisfy the dominant account. To do this, the solver:

- Determines if the candidate is an instream flow account. If so, it gives back all the remaining **Dominant Cutback** to the dominant account:

```
dominant appropriation = dominant appropriation + Dominant Cutback
Dominant Cutback = 0
```

and removes the candidate from the candidates list. Then it stops considering candidates and moves to the next downstream senior.

If the candidate is a diverter, the solver:

1. Finds the aggregate loss aggLoss between the dominant and the candidate,
2. Computes an amount to cut back the candidate, which is the lesser of what the candidate appropriates and what the dominant was cut back, adjusted for gain/loss:

```
candidateCutback = aggLoss *
                  min(candidate appropriation / aggLoss, Dominant Cutback)
```

3. Cuts back the candidate:

```
candidate appropriation = candidate appropriation - candidateCutback
```

4. Reduces the dominant's cutback by this amount adjusted for gain/loss:

```
Dominant Cutback = Dominant Cutback - candidateCutback / aggLoss
```

5. Gives the water back to the dominant account:

```
dominant appropriation = dominant appropriation + candidateCutback / aggLoss
```

6. If the candidate's appropriation is now zero, remove the candidate from the candidates list.

This process is repeated for every candidate until the `Dominant Cutback` is zero or there are no more candidates in the list.

Return {Slot Name,Value} or {Slot Name, DateTime, Value} List to Rule

After the solver has visited each water right in the subbasin in priority and computed their allocations and set the allocations on the cloned system, the solver creates a list of {slot name, value} pairs in the case of SolveWaterRights(), or {slot name, date-time, value} triplets in the case of SolveWaterRightsWithLags(), of allocations that are returned by rule function. The rule can access these pairs or triplets and set the slot equal to the value at the appropriate time.

Using Diagnostics with the Solvers

Because the solution to the water rights problem is computed by a computational subbasin, diagnostics pertinent to the solution are available by selecting the computational subbasin that appears as an argument to the predefined RPL function SolveWaterRights() or SolveWaterRightsWithLags().

This selection is made by opening the Diagnostics Manager (Utilities -> Diagnostics Manager), and from there editing settings for Rulebased Simulation. This opens a Rulebased Simulation Diagnostics Configuration dialog. In this dialog, select “show diagnostics for:” Run Management -> SimObj. In the same dialog, in a different pane, choose the objects to which these diagnostics apply; this pane is called “Show O diagnostics for”, and here you click on the button marked “Select”, which opens an object chooser dialog called Select Diagnostics Setting Object. Here you choose the “CompObj” with the name of the computational subbasin that is used for the solution. Be sure to click “Apply” or “OK” in all the dialogs, and be sure to give a destination for your diagnostics in the Diagnostics Manager dialog.

Note: *Turning on diagnostics for the solver produces voluminous output. You should be sure to restrict the date range for diagnostics as much as possible, and it would be best to send diagnostics to a file rather than to the diagnostics output window.*

Appendix A
Basin Specific Object Level Accounting Methods (OLAMs)

This document describes object level accounting methods specific to the Rio Grande model. In general, each account in this model has a water type of either “RioGrande” or “SanJuan.” In addition, there is often an account on the object name “RioGrande”. The categories and methods in the remainder of this document describe the logic used to allocation water to Slot Inflow and Gain Loss to these typed or named accounts.

This chapter covers the following topics:

Confluence Account Slot Inflow Category

The confluence has the following Rio Grande specific Slot Inflow method.

Rio Grande Inflow 2

This method sets the Slot Inflow for accounts with water type "RioGrande" proportionally to the Inflow to the account. All Slot Inflows on accounts with water type "SanJuan" are set to 0.0

The logic is follows:

```
Get a list of all the Inflow slots on accounts with water type RioGrande.

Register each as a RPL dependency.

Get a list of all the Slot Inflows slots on accounts with water type RioGrande.

RioGrandeInflowSUM = Sum of RioGrande Inflows

If the RioGrandeInflowSUM is zero, then divide Inflow 2 equally to all Rio Grande
accounts.

Otherwise, for each account of water type RioGrande:
```

$$\text{account.Slot Inflow} = \frac{\text{account.Inflow}}{\text{RioGrandeInflowSUM}} \times \text{Confluence.Inflow2}$$

```
Get a list of all the Slot Inflow slots on accounts with water type SanJuan

Set all SanJuan Slot Inflows to 0.0.
```

Reach Account Slot Inflow Category

The reach has the following Rio Grande-specific Slot Inflow methods:

- "Reconcile Rio Grande Outflow."
- "Rio Grande Local Inflow."

Reconcile Rio Grande Outflow

This method computes the total accounting Slot Inflow to reconcile the physical and accounting flows based on the water type of the accounts. Accounts with water type "SanJuan" have their Slot Inflows set to 0.0cfs. The remaining total reconciliation water is then proportioned to the Slot Inflow on accounts with the RioGrande water type. The logic is as follows:

```
Get a list of all the Inflow slots on accounts with water type RioGrande. and
register each as a dependency

RioGrandeInflowSUM = Sum of RioGrande Inflows

Get a list of all the Slot Inflows slots on accounts with water type RioGrande.

Get a list of all the Slot Inflow slots on accounts with water type SanJuan.

Get a list of all the Outflow slots on accounts with water type SanJuan and reg-
ister each as a dependency

SanJuanOutflowSUM = Sum of SanJuan Outflows

Get the Reach.Outflow (this includes the physical loss)
```

$$\text{RioGrandeSlotInflowTOTAL} = \text{Reach.Outflow} - \text{SanJuanOutflowSUM} - \text{RioGrandeInflowSUM}$$

```
If the RioGrandeInflowSUM is zero, then divide any Slot Inflows equally to all
Rio Grande accounts.

Otherwise, for each account with water type RioGrande:
```

$$\text{account.Slot Inflow} = \frac{\text{account.Inflow}}{\text{RioGrandeInflowSUM}} \times \text{RioGrandeSlotInflowTOTAL}$$

```
Set Slot Inflows on accounts with water type SanJuan to 0.0.
```

Rio Grande Local Inflow

This method sets all Local Inflow to the Slot Inflow on the account name "RioGrande." All accounts with Water type SanJuan are given zero Slot Inflows.

Following is the logic to compute the Slot Inflow on the Rio Grande account.

```
Get the single passthrough account named "RioGrande"

Get the RioGrande Slot Inflow slot

Set the RioGrande.Slot Inflow = Local Inflow (if Local Inflow is in use and valid)

Get a list of all the SanJuan Slot Inflow slots
```

```
Set all SanJuan Slot Inflows to 0.0.
```

Reservoir Account Gain Loss Category

The primary purpose of methods in this category is to set the Gain Loss slots on reservoir accounts. They also set values on additional slots as described within each method description. The following methods are described in this section:

- "Abiquiu, Cochiti and Jemez Gain Loss."
- "Elephant Butte and El Vado Gain Loss" on page 228.
- "Nambe Falls Gain Loss" on page 244.
- "Heron Gain Loss" on page 257.

Abiquiu, Cochiti and Jemez Gain Loss

Abiquiu, Cochiti and Jemez reservoirs each have their own Gain Loss Calculation method; however, all three methods function the same, as described below.

These methods are currently configured to function correctly for daily and monthly timesteps only. In order for these methods to solve for Account Gain Loss, values such as Surface Area, Evaporation and Precipitation Volume must be know (solved for by simulation); therefore, the execution time must be set to After Simulation.

These methods calculate the total San Juan losses as the difference between the Present Condition Loss and the Hypothetical Condition Loss. The Hypothetical Condition represents the state of the reservoir assuming only Rio Grande water is present, with no San Juan water. The total San Juan loss is then distributed to all San Juan accounts proportional to the storage in each account at the end of the previous timestep. This allows the San Juan Storages to solve at the current timestep.

$$totalSanJuanLosses = -(presentLoss - hypotheticalLoss)$$

$$\boldsymbol{Account}\textbf{.Gain Loss} = \frac{Account.\text{Storage(t-1)}}{totalPrevSanJuanStorage} \times totalSanJuanLosses$$

All other loss goes to the Rio Grande accounts. The temporary total Rio Grande loss is calculated as the difference between the Hypothetical Condition Loss and the Pre Reservoir Condition Loss. If the reservoir has a RioGrandeConservation account, and the account has water, then it takes a fraction of the loss equal to its percentage of total Rio Grande water in the reservoir. All remaining loss goes to the RioGrande account.

$$totalRioGrandeLossTemp = -(hypotheticalLoss - preResLoss)$$

$$\textbf{RioGrandeConservation.Gain Loss} = totalRioGrandeGainLossTemp \times \left(\frac{\text{RioGrandeConservation.Storage(t-1)} + RGConservInVol - RGConservOutVol}{totalRioGrandeStorage + totalRioGrandeLossTemp}\right)$$

This allows RioGrandeConservation.Storage to solve if the reservoir has a RioGrandeConservation account. Then the Gain Loss for the RioGrande account can be calculated.

$$\textbf{RioGrande.Gain Loss} = (Reservoir.\text{Storage} - totalSanJuanStorageSum) - \text{RioGrande.Storage(t-1)} + (-\text{RioGrande.Inflow} - RGTransfersInSum + \text{RioGrande.Outflow} + RGTransfersOutSum) \times Timestep - \text{RioGrandeConservation.Storage} - \text{RioGrandeConservation.Transfers In} \times Timestep - \text{Accumulated Permanent Sediment}$$

The details of the individual components are provided in "Slots specific to the Abiquiu, Cochiti or Jemez Gain Loss method" on page 206. The steps that each of these methods carry out are detailed in "Abiquiu, Cochiti or Jemez Gain Loss method steps" on page 210.

Slots specific to the Abiquiu, Cochiti or Jemez Gain Loss method

Selecting the Abiquiu, Cochiti or Jemez Gain Loss method instantiates the following slots:

+ Carryover Content

Type: Series Slot
Units: Volume
Description: The volume designated as Carryover storage
This value will only be non-zero if the Locked In slot has a value of 1 (reservoir is "locked in"). Values in this series slot are the same as the values in the Rio Grande Pools slot, Carryover Content column. If the initial timestep value on this slot is not known, but the value on the Rio Grande Pools slot Carryover Content column t is known, the table series slot value is copied to the series slot at the initial timestep and set as an input.
I/O: Initial value Input, run timesteps are Output
Supply Links: No

+ Conservation Gain Switch

Type: Series Slot
Units: None
Description: A binary switch indicating whether any gains at the timestep can be allocated to the RioGrandeConservation account.
This slot value must be either 1 (gains can be allocated to RioGrandeConservation) or 0 (no gains allocated to RioGrandeConservation). If this value is not input or set by a rule, it defaults to zero
I/O: Optional Input
Supply Links: No

+ Control and PreRes Areas

Type: Table Slot
Units: Area, Area, Area, Area, Area, Area
Description: The table contains a one row and six columns, one for each of the following: Control Area, Pre-Res Barren Area, Pre-Res Irrigated Area, Pre-Res Meadow Area, Pre-Res River Area, Pre-Res Lake Area
The areas in this table are used for calculating losses for pre-reservoir conditions.

I/O: Required Input
Supply Links: No

+ Effective Precip

Type: Series Slot
Units: Volume
Description: The effective precipitation calculated for the given timestep.
The total accumulated precipitation depth for the month is used to calculate the accumulated effective precipitation for the current day and the previous day. The difference is the Effective Precip for the current day. For a monthly timestep, the effective precipitation is calculated from the total precipitation for the current timestep.
I/O: Output
Supply Links: No

+ Effective Precipitation Table

Type: Table Slot
Units: Length, Fraction, Length
Description: The relationship between total accumulated precipitation depth for the month and the amount considered effective precipitation.
The first column gives a depth of monthly precipitation. The second column gives the fraction of that precipitation depth that is considered effective precipitation. The third column contains the accumulated effective precipitation value corresponding to the precipitation in the first column. Linear interpolation is used to calculate the effective precipitation depth based on the total precipitation depth for the month. Only the first and third columns are actually used in the interpolation. The second column is only for reference.
I/O: Required Input
Supply Links: No

+ Hold Pool

Type: Series Slot
Units: Volume
Description: The sum of the total San Juan storages, Carryover Content and Accumulated Permanent Sediment
This value is calculated at the end of the accounting method after all account storage values are known.
I/O: Output
Supply Links: No

+ Hypothetical Lake Elev River Area Table

Type: Table Slot
Units: Length, Area

Description: Relationship of Hypothetical River Area to Hypothetical Lake Elevation
This table is used to calculate river area for hypothetical conditions
I/O: Required Input
Supply Links: No

+ Incidental Content

Type: Series Slot
Units: Volume
Description: All Rio Grande storage that is not designated as Carryover storage or Conservation storage.

I/O: Output
Supply Links: No

+ Locked In

Type: Series Slot
Units: None
Description: A binary switch indicating if the reservoir can have water designated for Carryover Content
This slot value must be either 1 (Locked In, can have Carryover storage) or 0 (no Carryover storage)
I/O: Required Input
Supply Links: No

+ Net Loss Adjustment

Type: Series Slot
Units: Volume
Description: The difference between the Physical Model Net Loss and the losses calculated by the accounting method
If this value is positive the physical model loss was greater than the accounting method loss
I/O: Output
Supply Links: No

+ Physical Model Net Loss

Type: Series Slot
Units: Volume
Description: The net loss from the simulation methods (Evaporation - Precipitation).
A positive value represents a net loss
I/O: Output
Supply Links: No

+ PreRes Irrigated Area Loss Rate

Type:	Series Slot with Periodic Input
Units:	Velocity
Description:	The loss rate over irrigated area before the reservoir existed This value is used for calculating losses for pre-reservoir conditions.
I/O:	Required Input
Supply Links:	No

+ PreRes Meadow Area Loss Rate

Type:	Series Slot with Periodic Input
Units:	Velocity
Description:	The loss rate over meadow area before the reservoir existed This value is used for calculating losses for pre-reservoir conditions.
I/O:	Required Input
Supply Links:	No

+ Present Lake Elev River Area Table

Type:	Table Slot
Units:	Length, Area
Description:	Relationship of River Area to Pool Elevation This table is used to calculate river area for present conditions
I/O:	Required Input
Supply Links:	No

+ Volume Added to Carryover Storage

Type:	Series Slot
Units:	Volume
Description:	This volume is added to the Carryover Content at the timestep. If this value is not input or set by a rule, it defaults to zero.
I/O:	Optional Input
Supply Links:	No

The following table series slots are also set by this method. Details of each column value are given at the end of the method description. See:

- "Preliminary Data" on page 220.
- "PreReservoir Condition" on page 221.
- "Present Condition" on page 222.
- "Hypothetical Condition" on page 223.
- "Rio Grande Pools" on page 225.

- “Daily Reservoir Computations” on page 226.
- “Reservoir Summary” on page 227.

Abiquiu, Cochiti or Jemez Gain Loss method steps

All three methods (Abiquiu, Cochiti or Jemez Gain Loss method) carry out the following steps:

1. Check for previous Storage and Accrual on all accounts.
2. Call the selected Sediment Transport method to calculate Accumulated Perm Sediment.

Note: *See “Sediment Transport” on page 277 for a description of the Sediment Transport methods.*

---

3. Get lists of each type of account slot by water type.
4. Register account Inflow and Outflow slots as dependencies.
5. Calculate Effective Precipitation. See “Calculate Effective Precipitation” on page 211.
6. Calculate the evaporation depth and effective precipitation depth to use in loss calculations. See “Calculate the evaporation depth and effective precipitation depth to use in loss calculations” on page 212.
7. Calculate the individual components of the PreReservoir Condition Loss and the total PreRes Condition Loss. See “Calculate the individual components of the PreReservoir Condition Loss and the total PreRes Condition Loss” on page 212.
8. Calculate the individual components of the Present Condition Loss and the total Present Condition Loss. See“Calculate the individual components of the Present Condition Loss and the total Present Condition Loss” on page 212.
9. Sum the Rio Grande and San Juan previous storages and Rio Grande Transfers to use in loss calculations. See “Sum the Rio Grande and San Juan previous storages and Rio Grande Transfers to use in loss calculations” on page 213.
10. Set the seed value for the RioGrande storage iterative loop. See “Set the seed value for rioGrande Storage iterative loop” on page 213.
11. Carry out the Rio Grande iterative loop until the Rio Grande storage converges. See “Carry out the rioGrande Storage iterative loop until the Rio Grande storage converges” on page 213.
12. Calculate the Physical Model Net Loss and the Net Loss Adjustment. See “Calculate the Physical Model Net Loss and the Net Loss Adjustment” on page 215.
13. Calculate actual Rio Grande losses. See “Calculate actual Rio Grande losses” on page 215.
14. Adjust the temporary loss values, if necessary before setting final slot values. See“Adjust the temporary loss values, if necessary, according to the logic below before setting final slot values” on page 216.
15. Set the Gain Loss slots for the RioGrande and RioGrande “Set the Gain Loss slots for the RioGrande and RioGrande Conservation accounts” on page 220.
16. Set the remaining slots associated with this method. See “Set the remaining slots associated with the method” on page 220.

17. Set the summary values in the table series slots. See "Set summary values in table series slots" on page 220

Calculate Effective Precipitation

For a monthly timestep, this can be done based on the Precipitation Volume for the month, which will have been set in simulation. The Precipitation Volume calculation will already account for Ice Coverage if applicable. For a daily timestep, Effective Precipitation must be calculated using the cumulative sum of Precipitation Rate from the beginning of the month to the current day, and Ice Coverage is included explicitly. At a monthly timestep the average surface area is used, whereas the current (end of timestep) surface area is used for a daily timestep.

IF isMonthly

$$precipRate = \frac{\text{Precipitation Volume}}{(\text{Surface Area} + \text{Surface Area(t-1)})/2}$$

$$effectPrecipRate = TableInterpolation(\text{Effective Precipitaiton Table}, precipRate)$$

$$\text{Effective Precip} = effectPrecipRate \times \frac{\text{Surface Area} + \text{Surface Area(t-1)}}{2}$$

ELSE

$$prevCumulPrecip = \sum_{d = BegMonth}^{CurDay - 1} (\text{Precipitation Rate(d)} \times Timestep$$

$$\times (1 - \text{Surface Ice Coverage(d)}))$$

$$cumulPrecip = \sum_{d = BegMonth}^{CurDay} (\text{Precipitation Rate(d)} \times Timestep$$

$$\times (1 - \text{Surface Ice Coverage(d)}))$$

$$prevEffectPrecip = TableInterpolation(\text{Effective Precipitaiton Table}, prevCumulPrecip)$$

$$presentEffectPrecip = TableInterpolation(\text{Effective Precipitaiton Table}, cumulPrecip)$$

$$effectPrecipRate = presentEffectPrecip - prevEffectPrecip$$

$$\text{Effective Precip} = effectPrecipRate \times \text{Surface Area}$$

END IF

Calculate the evaporation depth and effective precipitation depth to use in loss calculations

For all following calculations, Area values for a daily timestep are current (end of timestep) areas. For a monthly timestep, Area values are averages of the end of the previous timestep and end of the current timestep values, similarly for Ice Coverage. If Ice Coverage is not applicable based on the selected evaporation method, it is zero in the calculations below.

$$evapDepth = \frac{\text{Evaporation}}{\text{Surface Area}}$$

$$effectivePrecipDepth = \frac{\text{Effective Precip}}{\text{Surface Area} \times (1 - \text{Surface Ice Coverage})}$$

If the Surface Area is zero, then *evapDepth* and *effecitvePrecipDepth* are zero, and if Ice Coverage is one, then *effectivePrecipDepth* is zero.

Calculate the individual components of the PreReservoir Condition Loss and the total PreRes Condition Loss

Each of these values is positive for a loss. The PreRes areas are contained in the Control and PreRes Areas table slot. The results of these calculations are set in the PreReservoir Condition table series slot.

$$preResBarrenLoss = preResBarrenArea \times effectPrecipDepth$$

$$preResIrrigLoss = preResIrrigArea \times \text{PreRes Irrigated Area Loss Rate} \times (1 - \text{Surface Ice Coverage}) \times Timestep$$

$$preResMeadowLoss = preResMeadowArea \times \text{PreRes Meadow Area Loss Rate} \times (1 - \text{Surface Ice Coverage}) \times Timestep$$

$$preResRiverLoss = preResRiverArea \times evapDepth$$

$$preResLoss = preResBarrenLoss + preResIrrigLoss + preResMeadowLoss + preResRiverLoss$$

Calculate the individual components of the Present Condition Loss and the total Present Condition Loss

Each of these values is positive for a loss. The results of these calculations are set in the Present Condition table series slot.

$$presRiverArea = TableInterpolation(\text{Present Lake Elev River Area Table, Pool Elevation})$$

$$presBarrenArea = totalArea - presRiverArea - \text{Surface Area}$$

$$presBarrenLoss = presBarrenArea \times effectPrecipDepth$$

$$presWaterSurfaceLoss = (\text{Surface Area} + presRiverArea) \times evapDepth$$

$$netInflowVolume = \text{Storage} - \text{Storage(t-1)} + \text{Outflow} \times Timestep$$
$$+ \text{Evaporation} - \text{Precipitation Volume} - \text{Storage Adjustment from Elev Vol Table Change}$$

The *netInflowVolume* will be negative if there is unidentified loss. If *netInflowVolume* is positive, then unidentified loss is zero. The Storage Adjustment from Elev Vol Table Change component is only included if the Time Varying Elevation Volume method is selected for the Sediment category.

$$unidentifiedLoss = Max(0, -netInflowVolume)$$

unidentifiedLoss will be positive for a loss.

$$presentLoss = presBarrenLoss + presWaterSurfaceLoss + unidentifiedLoss$$

Sum the Rio Grande and San Juan previous storages and Rio Grande Transfers to use in loss calculations

$$rioGrandePrevStorSum = \sum rioGrandeStorages(t-1)$$
$$sanJuanPrevStorSum = \sum sanJuanStorages(t-1)$$
$$sanJuanPsThruPrevStorSum = \sum sanJuanPassThruStorages(t-1)$$

$$totalPrevSanJuanStorage = sanJuanPrevStorSum + sanJuanPsThruPrevStorSum$$

rioGrandeStorages, *sanJuanStorages*, and *sanJuanPassThruStorages* are lists of storage slots on all accounts with the specified water type.

$$RGTansfersInSum = \sum rioGrandeTransfersIn$$

$$RGTransfersOutSum = \sum rioGrandeTransfersOut$$

rioGrandeTransfersIn and *rioGrandeTransersOut* are lists of Transfer slots on all accounts with the RioGrande water type.

Set the seed value for rioGrande Storage iterative loop

$$totalRioGrandeStorage = rioGrandePrevStorSum$$

Carry out the rioGrande Storage iterative loop until the Rio Grande storage converges

$$prevTotalRioGrandeStorage = totalRioGrandeStorage$$

To carry out Carry out the rioGrande Storage iterative loop until the Rio Grande storage converges, you must do the following in the order indicated:

- Calculate the Hypothetical Condition Loss. See “Calculate the Hypothetical Condition Loss” on page 214.
- Distribute San Juan gains/losses. See “Distribute San Juan gains/losses” on page 214.

- Test for convergence. See "Test for convergence" on page 215.

Calculate the Hypothetical Condition Loss

This is the loss assuming only Rio Grande water in the reservoir, no San Juan water. These results are set in the Hypothetical Condition table series slot. Losses here are positive values.

$$hypotheticalLakeElev = TableInterpolation(\text{Elevation Volume Table}, totalRGStorage)$$

$$hypotheticalSurfaceArea = TableInterpolation(\text{Elevation Area Table}, hypotheticalLakeElev)$$

$$hypotheticalRiverArea = TableInterpolation(\text{Hypothetical Lake Elev River Area Table}, hypothetcialLakeElev)$$

$$hypotheticalBarrenArea = totalArea - hypotheticalRiverArea - hypotheticalSurfaceArea$$

$$hypotheticalBarrenLoss = hypotheticalBarrenArea \times effectivePrecipDepth$$

$$hypotheticalWaterSurfaceLoss = (hypotheticalSurfaceArea + hypotheticalRiverArea) \times evapDepth$$

$$hypotheticalLoss = hypotheticalBarrenLoss + hypotheticalWaterSurfaceLoss$$

Distribute San Juan gains/losses

The total Gain/Loss allocated to San Juan storage accounts is the difference between the Present Condition Loss and the Hypothetical Condition Loss. A positive value for each of these components represents a loss. In the Gain Loss slot on each account, a positive value represents a gain, and a negative value represents a loss. Thus the negative of the difference is taken before distributing the loss to the Gain Loss slot. The loss is distributed based on the contributing percentage of the total storage at the end of the previous timestep. If the total San Juan storage at the end of the previous timestep was zero, then the Gain Loss is zero for all accounts.

$$totalSanJuanLosses = -(presentLoss - hypotheticalLoss)$$

$$Account.\text{Gain Loss} = \frac{Account.\text{Storage(t-1)}}{totalPrevSanJuanStorage} \times totalSanJuanLosses$$

for all San Juan accounts

This allows the San Juan accounts to solve for Storage. The sum of all San Juan storages can then be used to calculate the total Rio Grande storage.

$$totalSanJuanStorageSum = \sum sanJuanStorages + \sum sanJuanPassThruStorages$$

$$totalRioGrandeStorage = Reservoir.\text{Storage} - totalSanJuanStorageSum - AccumPermSediment$$

The *AccumPermSediment* is calculated by the selected method in the Sediment Transport category, and the value is set in the Est Sed Deposition table series slot.

Test for convergence

$$\varepsilon = \frac{prevTotalRioGrandeStorage - totalRioGrandeStorage}{totalRioGrandeStorage \times 1000}$$

If *totalRioGrandeStorage* is zero, then *prevTotalRioGrandeStorage* is used in the denominator. If both are zero, then epsilon is zero. The iterative loop repeats until the absolute value of epsilon is less than the specified tolerance in the Convergence Percentage slot or the maximum number of iterations is reached.

Calculate the Physical Model Net Loss and the Net Loss Adjustment

---

Note: *These values are used for reporting only. They are not used in later calculations.*

$$precip = \text{Precipitation Rate} \times Timestep \times \text{Surface Area}$$

$$\text{Physical Model Net Loss} = \text{Evaporation} - precip$$

A positive value for Physical Model Net Loss represents a loss.

$$totalSanJuanGainLoss = \sum sanJuanGainLosses + \sum sanJuanPassThruGainLosses$$

A negative value for *totalSanJuanGainLoss* represents a loss.

$$totalRioGrandeLossTemp = hypotheticalLoss - preResLoss$$

A positive value for *totalRioGrandeLossTemp* represents a loss.

$$\text{Net Loss Adjustment} = \text{Physical Model Net Loss} + totalSanJuanGainLoss - totalRioGrandeLossTemp$$

Calculate actual Rio Grande losses

If there is a RioGrandeConservation account, then the Rio Grande Losses are divided proportionally between the RioGrande account and the RioGrandeConservation account (details below). If there is no RioGrandeConservation account, then all RioGrandeConservation values in the steps below are zero.

$$RGConservInVol = (\text{RioGrandeConservation.Inflow} + \text{RioGrandeConservation.Transfers In}) \times Timestep$$

$$RGConservOutVol = (\text{RioGrandeConservation.Out} + \text{RioGrandeConservation.Transfers Out}) \times Timestep$$

$$RGConservLossTemp = totalRioGrandeLossTemp \times \left(\frac{\text{RioGrandeConservation.Storage(t-1)} + RGConservInVol - RGConservOutVol}{totalRioGrandeStorage + totalRioGrandeLossTemp}\right)$$

IF $totalRioGrandeStorage + totalRioGrandeLossTemp = 0$, then $RGConservLossTemp = 0$.

$$RGConservTemp = \text{RioGrandeConservation.Storage(t-1)} + RGConservInVol - RGConservOutVol - RGConservLossTemp$$

If the reservoir is "Locked In" (indicated by a value of 1 in the Locked In slot), a percentage of the Rio Grande loss is allocated to the Carryover Content, proportional to the percent contribution to total Rio Grande storage. If the reservoir is not locked in, all Carryover values are zero. Note that Carryover Content is not a slot on the accounts but is a separate slot **Carryover Content** on the reservoir object.

IF $\text{Locked In} = 0$

$$RGCarryOverLossTemp = 0$$

$$RGCarryOverTemp = 0$$

ELSE

IF $totalRioGrandeStorage + totalRioGrandeLossTemp = 0$

$$RGCarryOverLossTemp = 0$$

ELSE

$$RGCarryOverLossTemp = totalRioGrandeLossTemp \times \left(\frac{\text{Carryover Content(t-1) + Volume Added}}{totalRioGrandeStorage + totalRioGrandeLossTemp}\right)$$

$$RGCarryOverTemp = \text{Carryover Content(t-1) + Volume Added} - RGCarryOverLossTemp$$

END IF

END IF

All other Rio Grande losses and storage go to the Incidental pool.

$$RGIncidentalTemp = totalRioGrandeStorage - RGConservTemp - RGCarryOverTemp$$

$$RGIncidentalLossTemp = totalRioGrandeLossTemp - RGCarryOverLossTemp - RGConservLossTemp$$

Adjust the temporary loss values, if necessary, according to the logic below before setting final slot values

If *totalRioGrandeLossTemp* is negative, representing a gain, then the gain is either divided proportionally between the Conservation and Incidental pools (if the Conservation Gain Switch is "On"), or the entire gain goes to the Incidental Pool.

IF $totalRioGrandeLossTemp < 0$

$$RGCarryOverLossFinal = 0$$

IF $(\text{Conservation Gain Switch} = 1)$ AND $RGConservTemp > 0$

IF $RGIncidentalTemp \le 0$

$$RGIncidentalLossFinal = 0$$

$$RGConservLossFinal = totalRioGrandeLossTemp$$

ELSE

$$RGIncidentalLossFinal = \frac{RGIncidentalLossTemp}{\left(\frac{RGConservTemp + RGIncidentalTemp}{totalRioGrandeStorage}\right)}$$

$$RGConservLossFinal = \frac{RGConservLossTemp}{\left(\frac{RGConservTemp + RGIncidentalTemp}{totalRioGrandeStorage}\right)}$$

END IF

ELSE

$$RGIncidentalLossFinal = totalRioGrandeLossTemp$$

$$RGConservLossFinal = 0$$

END IF

ELSE

For the remaining cases, *totalRioGrandeLossTemp* is positive (a loss). The temporary loss can be adjusted based on which pools have water.

IF$(RGIncidentalTemp > 0)$ AND$(RGCarryOverTemp \le 0)$ AND$(RGConservTemp \le 0)$

$$RGIncidentalLossFinal = totalRioGrandeLossTemp$$

$$RGCarryOverLossFinal = 0$$

$$RGConservLossFinal = 0$$

ELSE IF $(RGIncidentalTemp > 0)$ AND $(RGCarryOverTemp > 0)$

AND$(RGConservTemp \le 0)$

$$RGIncidentalLossFinal = \frac{RGIncidentalLossTemp}{\left(\frac{RGCarryOverTemp + RGIncidentalTemp}{totalRioGrandeStorage}\right)}$$

$$RGCarryOverLossFinal = \frac{RGCarryOverLossTemp}{\left(\frac{RGCarryOverTemp + RGIncidentalTemp}{totalRioGrandeStorage}\right)}$$

$$RGConservLossFinal = 0$$

ELSE IF $(RGIncidentalTemp > 0)$ AND $(RGCarryOverTemp \le 0)$

AND $(RGConservTemp > 0)$

$$RGIncidentalLossFinal = \frac{RGIncidentalLossTemp}{\left(\frac{RGCarryOverTemp + RGIncidentalTemp}{totalRioGrandeStorage}\right)}$$

$$RGCarryOverLossFinal = 0$$

$$RGConservLossFinal = \frac{RGConservLossTemp}{\left(\frac{RGConservTemp + RGIncidentalTemp}{totalRioGrandeStorage}\right)}$$

ELSE IF $(RGIncidentalTemp \le 0)$ AND $(RGCarryOverTemp > 0)$

AND $(RGConservTemp > 0)$

$$RGIncidentalLossFinal = 0$$

$$RGCarryOverLossFinal = \frac{RGCarryOverLossTemp}{\left(\frac{RGCarryOverTemp + RGConservTemp}{totalRioGrandeStorage}\right)}$$

$$RGConservLossFinal = \frac{RGConservLossTemp}{\left(\frac{RGCarryOverTemp + RGConservTemp}{totalRioGrandeStorage}\right)}$$

ELSE IF $(RGIncidentalTemp \le 0)$ AND $(RGCarryOverTemp > 0)$

AND $(RGConservTemp \le 0)$

$$RGIncidentalLossFinal = 0$$

$$RGCarryOverLossFinal = totalRioGrandeLossTemp$$

$$RGConservLossFinal = 0$$

ELSE IF $(RGIncidentalTemp \le 0)$ AND $(RGCarryOverTemp \le 0)$

AND $(RGConservTemp > 0)$

$$RGIncidentalLossFinal = 0$$

$$RGCarryOverLossFinal = 0$$

$$RGConservLossFinal = totalRioGrandeLossTemp$$

ELSE IF $(RGIncidentalTemp \le 0)$ AND $(RGCarryOverTemp \le 0)$

AND $(RGConservTemp \le 0)$

$RGIncidentalLossFinal = totalRioGrandeLossTemp$

$RGCarryOverLossFinal = 0$

$RGConservLossFinal = 0$

ELSE (all are positive)

$RGIncidentalLossFinal = RGIncidentalLossTemp$

$RGCarryOverLossFinal = RGCarryOverLossTemp$

$RGConservLossFinal = RGConservationLossTemp$

END IF

END IF

$$RGConservation = \text{RioGrandeConservation.Storage(t-1)} + (\text{RioGrandeConservation.Inflow} + \text{RioGrandeConservation.Transfers In} - \text{RioGrandeConservation.Outflow} - \text{RioGrandeConservation.Transfers Out}) \times Timestep - RGConservationLossFinal$$

If it is the first timestep locked in, then Carryover gets all of the storage and the Incidental pool gets all of the loss.

IF $\text{Locked In} = 1$

IF $\text{Locked In(t-1)} = 0$

$RGIncidental = 0$

$RGIncindentalLossFinal = RGCarryOverLossFinal + RGIncidentalLossFinal$

$RGCarryOver = RGCarryOverTemp + RGIncidentalTemp$

$RGCarryOverLossFinal = 0$

ELSE (not the first day locked in)

$RGCarryOver = \text{Carryover Content(t-1)} + \text{Volume Added} - RGCarryOverLossFinal$

$RGIncidental = totalRioGrandeStorage - RGCarryOver - RGConservation$

END IF

ELSE (The reservoir is not locked in)

$RGCarryOver = 0$

$RGIncidental = totalRGStorage - RGCarryOver - RGConservation$

END IF

Set the Gain Loss slots for the RioGrande and RioGrande Conservation accounts

$$\text{RioGrande.Gain Loss} = \\ Reservoir\text{.Storage} - totalSanJuanStorageSum - \text{RioGrande.Storage(t-1)} \\ + (-\text{RioGrande.Inflow} - RGTransfersInSum \\ + \text{RioGrande.Outflow} + RGTransfersOutSum) \times Timestep \\ - RGConservation + \text{RioGrandeConservation.Transfers In} \times Timestep \\ - AccumPermSed$$

$$\text{RioGrandeConservation.Gain Loss} = -RGConservLossFinal$$

Set the remaining slots associated with the method

$$\text{Carryover Content} = RGCarryOver$$

$$\text{Incidental Content} = RGIncidental$$

$$\text{Hold Pool} = totalSanJuanStorageSum + RGCarryOver + AccumPermSed$$

Set summary values in table series slots

Each table below represents a table series slot. The row names in the following tables represent the column headers in the table series slots.

Preliminary Data

Column	Value
Elevation	*Reservoir*.Pool Elevation
Content	*Reservoir*.Storage
Area	*Reservoir*.Surface Area
Evaporation	*Reservoir*.Evaporation

Column	Value
Precipitation	*Reservoir*.Precipitation Volume
Unidentified Loss	$unidentifiedLoss = Max(0, -(\text{Storage} - \text{Storage(t-1)} + \text{Outflow} \times Timestep$ $+ \text{Evaporation} - \text{Precipitation Volume}$ $- \text{Storage Adjustment from Elev Vol Table Change}))$
Outflow (cfs)	*Reservoir*.Outflow
Outflow (af/d)	*Reservoir*.Outflow
Inflow (cfs)	*Reservoir*.Evaporation
Inflow (af/d)	*Reservoir*.Evaporation

PreReservoir Condition

Column	Value
River Channel Area	Control and PreRes Areas(Pre-Res River Area)
Barren Area	Control and PreRes Areas(Pre-Res Barren Area)
Meadow and Town Area	Control and PreRes Areas(Pre-Res Meadow Area)

Column	Value
River Area Loss	$preResRiverLoss = preResRiverArea \times evapDepth$
Barren Area Loss	$preResBarrenLoss = preResBarrenArea \times effectPrecipDepth$
Irrigated Area Loss	$preResIrrigLoss = preResIrrigArea \times \text{PreRes Irrigated Area Loss Rate} \times (1 - \text{Surface Ice Coverage}) \times Timestep$
Meadow and Town Area Loss	$preResMeadowLoss = preResMeadowArea \times \text{PreRes Meadow Area Loss Rate} \times (1 - \text{Surface Ice Coverage}) \times Timestep$
PreRes Condition Loss	$preResLoss = preResBarrenLoss + preResIrrigLoss + preResMeadowLoss + preResRiverLoss$

Present Condition

Column	Value
Water Surface Elev	Pool Elevation
Lake Surface Area	Surface Area
River Channel Area	presRiverArea
Barren Area	presBarrenArea

Column	Value
Total Area	Control and PreRes Areas(Control Area)
Water Surface Area Loss	$presWaterSurfaceLoss = (\text{Surface Area} + presRiverArea) \times evapDepth$
Barren Area Loss	$presBarrenLoss = presBarrenArea \times effectPrecipDepth$
Present Condition Loss	$presentLoss = presBarrenLoss + presWaterSurfaceLoss$
PreRes Condition Loss	$preResLoss = preResBarrenLoss + preResIrrigLoss + preResMeadowLoss + preResRiverLoss$
Net RG Plus SJ-C Loss	$presentLoss - preResLoss$

Hypothetical Condition

Column	Value
Total Content	*Reservoir*.Storage
SJ-C Content	totalSanJuanStorageSum
RG Content	totalRioGrandeStorage

Column	Value
Lake Surface Area	hypotheticalSurfaceArea
River Channel Area	hypotheticalRiverArea
Barren Area	hypotheticalBarrenArea
Water Surface Area Loss	$hypotheticalWaterSurfaceLoss = (hypotheticalSurfaceArea + hypotheticalRiverArea) \times evapDepth$
Barren Area Loss	$hypotheticalBarrenLoss = hypotheticalBarrenArea \times effectivePrecipDepth$
Hypo. Condition Loss	$hypotheticalLoss = hypotheticalBarrenLoss + hypotheticalWaterSurfaceLoss$
PreRes Condition Loss	$preResLoss = preResBarrenLoss + preResIrrigLoss + preResMeadowLoss + preResRiverLoss$
Net RG Condition Loss	totalRioGrandeLossTemp
Net RG Plus SJ-C Loss	$presentLoss - preResLoss$
Net SJ-C Loss	$\sum -Account.\text{Gain Loss}$, for all accounts with SanJuan water type

Rio Grande Pools

Column	Value
Content Change	Storage – Storage(t-1)
Total Loss	$RGCarryOverLossFinal + RGIncidentalLossFinal + RGConservLossFinal$
Carryover Loss	RGCarryOverLossFinal
Incidental Loss	RGIncidentalLossFinal
Conservation Loss	RGConservationLossFinal
Release From Storage	$rioGrandePrevStorSum - totalRioGrandeStorage - totalRioGrandeLossTemp$
Total RG Content	$RGCarryOver + RGIncidental + RGConservation$
Carryover Content	RGCarryOver
Incidental Content	RGIncidental
Conservation Content	RGConservation

Daily Reservoir Computations

Column	Value
Water Surface Elev	*Reservoir*.Pool Elevation
Total Content	*Reservoir*.Storage
Change in Content	Storage – Storage(t-1)
RG Release (cfs)	$\sum Account.\text{Outflow}$, for all accounts with RioGrande water type
RG Release (af/d)	$\sum Account.\text{Outflow}$, for all accounts with RioGrande water type
SJ-C Release (cfs)	$\sum Account.\text{Outflow}$, for all accounts with SanJuan water type
SJ-C Release (af/d)	$\sum Account.\text{Outflow}$, for all accounts with SanJuan water type
Evap	*Reservoir*.Evaporation
Precip	*Reservoir*.Precipitation Volume
RG Inflow (cfs)	$\sum Account.\text{Inflow}$, for all accounts with RioGrande water type
RG Inflow (af/d)	$\sum Account.\text{Inflow}$, for all accounts with RioGrande water type

Column	Value
SJ-C Inflow (cfs)	$\sum Account.\text{Inflow}$, for all accounts with SanJuan water type
SJ-C Inflow (af/d)	$\sum Account.\text{Inflow}$, for all accounts with SanJuan water type

Reservoir Summary

Column	Value
Total Drawdown	Storage(t-1) – Storage – Storage Adjustment from Elev Vol Table Change
RG Loss	totalRioGrandeLossTemp
SJ-C Loss	$\sum -Account.\text{Gain Loss}$, for all accounts with SanJuan water type
Total Loss	SJ-C Loss + totalRioGrandeLossTemp
RG Release From Storage	$rioGrandePrevStorSum - totalRioGrandeStorage - totalRioGrandeLossTemp$
SJC Release From Storage	$totalPrevSanJuanStorage - totalSanJuanStorageSum + \sum AllSanJuanAccounts.\text{Gain Loss}$
RG Content	totalRioGrandeStorage

Column	Value
SJ-C Content	totalSanJuanStorageSum
Total Content	*Reservoir*.Storage

Elephant Butte and El Vado Gain Loss

Elephant Butte and El Vado reservoirs each have their own Gain Loss Calculation method. The Elephant Butte Loss with RG Compact method can also be selected. All three methods function the same, with a few small exceptions, as described below.

These methods are currently configured to function correctly for daily and monthly timesteps only. In order for these methods to solve for Account Gain Loss, values such as Surface Area, Evaporation and Precipitation Volume must be know (solved for by simulation); therefore, the execution time must be set to After Simulation.

These methods calculate the total San Juan losses as the difference between the Present Condition Loss and the Hypothetical Condition Loss. The Hypothetical Condition represents the state of the reservoir assuming only Rio Grande water is present, with no San Juan water. The total San Juan loss is then distributed to all San Juan accounts proportional to the storage in each account at the end of the previous timestep. This allows the San Juan Storages to solve at the current timestep.

$$totalSanJuanLosses = -(presentLoss - hypotheticalLoss)$$

$$\boldsymbol{Account}\textbf{.Gain Loss} = \frac{Account\text{.Storage(t-1)}}{totalPrevSanJuanStorage} \times totalSanJuanLosses, \text{ for all San Juan accounts}$$

All other loss goes to the Rio Grande accounts. The total Rio Grande loss is then distributed to all Rio Grande accounts proportional to the storage in each account at the end of the previous timestep.

$$\begin{aligned} totalRioGrandeLoss = {} & (Reservoir\text{.Storage} - totalSanJuanStorageSum) \\ & - rioGrandePrevStorSum + \\ & (-RGInflowSum \\ & \ - RGTansfersInSum \\ & \ + RGOutflowSum \\ & \ + RGTransfersOutSum) \times Timestep \\ & - \text{Storage Adjustment from Elev Vol Table Change} \end{aligned}$$

$$\boldsymbol{Account}\textbf{.Gain Loss} = \frac{Account\text{.Storage(t-1)}}{rioGrandePrevStorSum} \times totalRioGrandeLoss\text{, for all Rio Grande accounts}$$

The details of the individual components are provided in “Slots specific to the Elephant Butte and El Vado Gain Loss method” on page 229. The steps carried out by each method are detailed in “Steps carried out by the Elephant Butte and El Vado Gain Loss method” on page 231.

Slots specific to the Elephant Butte and El Vado Gain Loss method

Selecting the Elephant Butte Gain Loss, El Vado Gain Loss or the Elephant Butte Loss with RG Compact method instantiates the following slots:

Control and PreRes Areas

Type: Table Slot

Units: Area, Area, Area, Area, Area, Area

Description: The table contains a one row and six columns, one for each of the following: Control Area, Pre-Res Barren Area, Pre-Res Irrigated Area, Pre-Res Meadow Area, Pre-Res River Area, Pre-Res Lake Area

The areas in this table are used for calculating losses for pre-reservoir conditions.

I/O: Required Input

Supply Links: No

Effective Precip

Type: Series Slot

Units: Volume

Description: The effective precipitation calculated for the given timestep.

The total accumulated precipitation depth for the month is used to calculate the accumulated effective precipitation for the current day and the previous day. The difference is the Effective Precip for the current day. For a monthly timestep, the effective precipitation is calculated from the total precipitation for the current timestep.

I/O: Output

Supply Links: No

Effective Precipitation Table

Type: Table Slot

Units: Length, Fraction, Length

Description: The relationship between total accumulated precipitation depth for the month and the amount considered effective precipitation.

The first column gives a depth of monthly precipitation. The second column gives the fraction of that precipitation depth that is considered effective precipitation. The third column contains the accumulated effective precipitation value corresponding to the precipitation in the first column. Linear interpolation is used to calculate the effective precipitation depth based on the total precipitation depth for the month. Only the first and third columns are actually used in the interpolation. The second column is only for reference.

I/O: Required Input

Supply Links: No

Hypothetical Lake Elev River Area Table

Type: Table Slot

Units: Length, Area
Description: Relationship of Hypothetical River Area to Hypothetical Lake Elevation
This table is used to calculate river area for hypothetical conditions
I/O: Required Input
Supply Links: No

+ Net Loss Adjustment

Type: Series Slot
Units: Volume
Description: The difference between the Physical Model Net Loss and the losses calculated by the accounting method
If this value is positive the physical model loss was greater than the accounting method loss
I/O: Output
Supply Links: No

+ Physical Model Net Loss

Type: Series Slot
Units: Volume
Description: The net loss from the simulation methods (Evaporation - Precipitation).
A positive value represents a net loss
I/O: Output
Supply Links: No

+ PreRes Irrigated Area Loss Rate

Type: Series Slot with Periodic Input
Units: Velocity
Description: The loss rate over irrigated area before the reservoir existed
This value is used for calculating losses for pre-reservoir conditions.
I/O: Required Input
Supply Links: No

+ PreRes Meadow Area Loss Rate

Type: Series Slot with Periodic Input
Units: Velocity
Description: The loss rate over meadow area before the reservoir existed
This value is used for calculating losses for pre-reservoir conditions.
I/O: Required Input
Supply Links: No

+ Present Lake Elev River Area Table

Type:	Table Slot
Units:	Length, Area
Description:	Relationship of River Area to Pool Elevation This table is used to calculate river area for present conditions
I/O:	Required Input
Supply Links:	No

The following table series slots are also set by this method. Details of each column value are given at the end of the method description.

- “Preliminary Data” on page 238
- “PreReservoir Condition” on page 239
- “Present Condition” on page 240
- “Hypothetical Condition” on page 241
- “Daily Reservoir Computations” on page 242
- “Reservoir Summary” on page 243

Steps carried out by the Elephant Butte and El Vado Gain Loss method

The Elephant Butte and El Vado Gain Loss method carries out the following steps:

1. Check for previous Storage and Accrual on all accounts.
2. Call the selected Sediment Transport method to calculate Accumulated Perm Sediment.
3. Get lists of each type of account slot by water type.
4. Register account Inflow and Outflow slots as dependencies.
5. Calculate effective precipitation. See “Calculate Effective Precipitation” on page 232.
6. Calculate the evaporation depth and effective precipitation depth to use in loss calculations. See “Calculate the evaporation depth and effective precipitation depth to use in loss calculations” on page 233.
7. Calculate the individual components of the PreReservoir Condition Loss and the total PreRes Condition Loss. See “Calculate the individual components of the PreReservoir Condition Loss and the total PreRes Condition Loss” on page 233.
8. Sum the Rio Grande and San Juan previous storages and Rio Grande Transfers to use in loss calculations. See “Sum the Rio Grande and San Juan previous storages and Rio Grande Transfers to use in loss calculations” on page 234.
9. Set the seed value for the RioGrande iterative loop. See “Set the seed value for the RioGrande Storage iterative loop” on page 234.
10. Carry out the RioGrande Storage iterative loop until the Rio Grande storage converges. See “Carry out the RioGrande Storage iterative loop until the Rio Grande storage converges” on page 235.

11. Calculate the Rio Grande gains/losses. See “Calculate the Rio Grande gains/losses” on page 236.
12. Distribute the Rio Grande gains/losses. See “Distribute the Rio Grande gains/losses” on page 236.
13. Allocate the Storage Adjustment from Elev Vol Table change to the RioGrande account. See “Allocate the Storage Adjustment from Elev Vol Table Change to the RioGrande account.” on page 237.
14. Calculate the physical model net loss and the net loss adjustment. See “Calculate the Physical Model Net Loss and the Net Loss Adjustment” on page 237.
15. Set summary values in the table series slots. See “Set summary values in table series slots” on page 237.

Calculate Effective Precipitation

For a monthly timestep, this can be done based on the Precipitation Volume for the month, which will have been set in simulation. The Precipitation Volume calculation will already account for Ice Coverage if applicable. For a daily timestep, Effective Precipitation must be calculated using the cumulative sum of Precipitation Rate from the beginning of the month to the current day, and Ice Coverage is included explicitly. At a monthly timestep the average surface area is used, whereas the current (end of timestep) surface area is used for a daily timestep.

IF isMonthly

$$precipRate = \frac{\text{Precipitation Volume}}{(\text{Surface Area} + \text{Surface Area(t-1)})/2}$$

$$effectPrecipRate = TableInterpolation(\text{Effective Precipitaiton Table}, precipRate)$$

$$\text{Effective Precip} = effectPrecipRate \times \frac{\text{Surface Area} + \text{Surface Area(t-1)}}{2}$$

ELSE

$$prevCumulPrecip = \sum_{d = BegMonth}^{CurDay - 1} (\text{Precipitation Rate(d)} \times Timestep$$

$$\times (1 - \text{Surface Ice Coverage(d)}))$$

$$cumulPrecip = \sum_{d = BegMonth}^{CurDay} (\text{Precipitation Rate(d)} \times Timestep$$

$$\times (1 - \text{Surface Ice Coverage(d)}))$$

$$prevEffectPrecip = TableInterpolation(\text{Effective Precipitation Table}, prevCumulPrecip)$$

$$presentEffectPrecip = TableInterpolation(\text{Effective Precipitation Table}, cumulPrecip)$$

$$effectPrecipRate = presentEffectPrecip - prevEffectPrecip$$

$$\text{Effective Precip} = effectPrecipRate \times \text{Surface Area}$$

END IF

Calculate the evaporation depth and effective precipitation depth to use in loss calculations

For all following calculations, Area values for a daily timestep are current (end of timestep) areas. For a monthly timestep, Area values are averages of the end of the previous timestep and end of the current timestep values, similarly for Ice Coverage. If Ice Coverage is not applicable based on the selected evaporation method, it is zero in the calculations below.

$$evapDepth = \frac{\text{Evaporation}}{\text{Surface Area}}$$

$$effectivePrecipDepth = \frac{\text{Effective Precip}}{\text{Surface Area} \times (1 - \text{Surface Ice Coverage})}$$

If the Surface Area is zero, then *evapDepth* and *effecitvePrecipDepth* are zero, and if Ice Coverage is one, then *effectivePrecipDepth* is zero.

Calculate the individual components of the PreReservoir Condition Loss and the total PreRes Condition Loss

Each of these values is positive for a loss. The PreRes areas are contained in the Control and PreRes Areas table slot. The results of these calculations are set in the PreReservoir Condition table series slot.

$$preResBarrenLoss = preResBarrenArea \times effectPrecipDepth$$

$$preResIrrigLoss =$$
$$preResIrrigArea \times \text{PreRes Irrigated Area Loss Rate} \times (1 - \text{Surface Ice Coverage}) \times Timestep$$

$$preResMeadowLoss =$$
$$preResMeadowArea \times \text{PreRes Meadow Area Loss Rate} \times (1 - \text{Surface Ice Coverage}) \times Timestep$$

$$preResRiverLoss = preResRiverArea \times evapDepth$$

$$preResLoss =$$
$$preResBarrenLoss + preResIrrigLoss + preResMeadowLoss + preResRiverLoss$$

- Calculate the individual components of the Present Condition Loss and the total Present Condition Loss.

Each of these values is positive for a loss. The results of these calculations are set in the Present Condition table series slot.

$$presRiverArea = TableInterpolation(\text{Present Lake Elev River Area Table, Pool Elevation})$$

$$presBarrenArea = totalArea - presRiverArea - \text{Surface Area}$$

$$presBarrenLoss = presBarrenArea \times effectPrecipDepth$$

$$presWaterSurfaceLoss = (\text{Surface Area} + presRiverArea) \times evapDepth$$

$$netInflowVolume = \text{Storage} - \text{Storage(t-1)} + \text{Outflow} \times Timestep$$
$$+ \text{Evaporation} - \text{Precipitation Volume} - \text{Storage Adjustment from Elev Vol Table Change}$$

The *netInflowVolume* will be negative if there is unidentified loss. If *netInflowVolume* is positive, then unidentified loss is zero. The Storage Adjustment from Elev Vol Table Change component is only included if the Time Varying Elevation Volume method is selected for the Sediment category.

$$unidentifiedLoss = Max(0, -netInflowVolume)$$

unidentifiedLoss will be positive for a loss.

IF *Reservoir* = Elephant Butte

$$presentLoss = presBarrenLoss + presWaterSurfaceLoss$$

ELSE

$$presentLoss = presBarrenLoss + presWaterSurfaceLoss + unidentifiedLoss$$

END IF

Sum the Rio Grande and San Juan previous storages and Rio Grande Transfers to use in loss calculations

$$rioGrandePrevStorSum = \sum rioGrandeStorages(t-1)$$
$$sanJuanPrevStorSum = \sum sanJuanStorages(t-1)$$
$$sanJuanPsThruPrevStorSum = \sum sanJuanPassThruStorages(t-1)$$

$$totalPrevSanJuanStorage = sanJuanPrevStorSum + sanJuanPsThruPrevStorSum$$

rioGrandeStorages, sanJuanStorages and *sanJuanPassThruStorages* are lists of storage slots on all accounts with the specified water type.

$$RGTansfersInSum = \sum rioGrandeTransfersIn$$

$$RGTransfersOutSum = \sum rioGrandeTransfersOut$$

rioGrandeTransfersIn and *rioGrandeTransersOut* are lists of Transfer slots on all accounts with the RioGrande water type.

Set the seed value for the RioGrande Storage iterative loop

$$totalRioGrandeStorage = rioGrandePrevStorSum$$

Carry out the RioGrande Storage iterative loop until the Rio Grande storage converges

$$prevTotalRioGrandeStorage = totalRioGrandeStorage$$

To carry out the RioGrande Storage iterative loop until the Rio Grande storage converges, you must do the following in the order indicated:

- Calculate the hypothetical condition loss. See “Calculate the Hypothetical Condition Loss”
- Calculate and distribute the San Jan gains/losses. See “Calculate and distribute San Juan gains/losses” on page 235.
- Test for convergence. See “Test for convergence” on page 236.

Calculate the Hypothetical Condition Loss

This is the loss assuming only Rio Grande water in the reservoir, no San Juan water. These results are set in the Hypothetical Condition table series slot. Losses here are positive values.

$$hypotheticalLakeElev = TableInterpolation(\text{Elevation Volume Table}, totalRGStorage)$$

$$hypotheticalSurfaceArea = TableInterpolation(\text{Elevation Area Table}, hypotheticalLakeElev)$$

$$hypotheticalRiverArea = TableInterpolation(\text{Hypothetical Lake Elev River Area Table}, hypothetcialLakeElev)$$

$$hypotheticalBarrenArea = totalArea - hypotheticalRiverArea - hypotheticalSurfaceArea$$

$$hypotheticalBarrenLoss = hypotheticalBarrenArea \times effectivePrecipDepth$$

$$hypotheticalWaterSurfaceLoss = (hypotheticalSurfaceArea + hypotheticalRiverArea) \times evapDepth$$

$$hypotheticalLoss = hypotheticalBarrenLoss + hypotheticalWaterSurfaceLoss$$

Calculate and distribute San Juan gains/losses

The total Gain/Loss allocated to San Juan storage accounts is the difference between the Present Condition Loss and the Hypothetical Condition Loss. A positive value for each of these components represents a loss. In the Gain Loss slot on each account, a positive value represents a gain, and a negative value represents a loss. Thus the negative of the difference is taken before distributing the loss to the Gain Loss slot. The loss is distributed based on the contributing percentage of the total storage at the end of the previous timestep. If the total San Juan storage at the end of the previous timestep was zero, then the Gain Loss is zero for all accounts.

$$totalSanJuanLosses = -(presentLoss - hypotheticalLoss)$$

$$Account.\text{Gain Loss} = \frac{Account.\text{Storage(t-1)}}{totalPrevSanJuanStorage} \times totalSanJuanLosses$$

for all San Juan accounts

This allows the San Juan accounts to solve for Storage. The sum of all San Juan storages can then be used to calculate the total Rio Grande storage.

$$totalSanJuanStorageSum = \sum sanJuanStorages + \sum sanJuanPassThruStorages$$

$$totalRioGrandeStorage = Reservoir.\text{Storage} - totalSanJuanStorageSum$$

Test for convergence

$$\varepsilon = \frac{prevTotalRioGrandeStorage - totalRioGrandeStorage}{totalRioGrandeStorage \times 1000}$$

If *totalRioGrandeStorage* is zero, then *prevTotalRioGrandeStorage* is used in the denominator. If both are zero, then epsilon is zero. The iterative loop repeats until the absolute value of epsilon is less than the specified tolerance in the Convergence Percentage slot or the maximum number of iterations (Max Iterations slot) is reached.

Calculate the Rio Grande gains/losses

The Storage Adjustment from Elev Vol Table Change is applicable only if the Time Varying Elevation Volume method is selected for the Sediment category. It is removed at this point in the calculation because all loss from the Elevation-Volume change will be allocated later to the RioGrande account.

$$\begin{aligned} totalRioGrandeLoss = {} & (Reservoir.\text{Storage} - totalSanJuanStorageSum) \\ & - rioGrandePrevStorSum + \\ & (-RGInflowSum \\ & \;-RGTansfersInSum \\ & \;+RGOutflowSum \\ & \;+RGTransfersOutSum) \times Timestep \\ & - \text{Storage Adjustment from Elev Vol Table Change} \end{aligned}$$

Distribute the Rio Grande gains/losses

The loss is distributed based on the contributing percentage of the total storage at the end of the previous timestep. If the selected method is Elephant Butte Loss with RG Compact, then the loss is distributed to all accounts with the RioGrande water type, even if they have zero or negative storage. If the selected method is Elephant Butte Gain Loss or El Vado Gain Loss, the gain/loss is only distributed to accounts with positive storage. If all accounts with the RioGrande *water type* have zero or negative storage at the previous timestep, then all of the gain/loss goes to the RioGrande *account*.

IF Loss Method = Elephant Butte Loss with RG Compact

$$Account.\text{Gain Loss} = \frac{Account.\text{Storage(t-1)}}{rioGrandePrevStorSum} \times totalRioGrandeLoss\ ,$$

for all Rio Grande accounts

ELSE IF (Loss Method = Elephant Butte Gain Loss)

OR (Loss Method = El Vado Gain Loss)

For all Rio Grande accounts with $Account$.Storage(t-1) > 0 :

$$positiveRGPrevStorSum = \sum Account.\text{Storage(t-1)}$$

$$Account.\text{Gain Loss} = \frac{Account.\text{Storage(t-1)}}{positiveRGPrevStorSum} \times totalRioGrandeLoss$$

IF $positiveRGPRevStorSum \le 0$

$$\text{RioGrande.Gain Loss} = totalRioGrandeLoss$$

END IF

END IF

Allocate the Storage Adjustment from Elev Vol Table Change to the RioGrande account.

This value will be negative for a loss.

$$\text{RioGrande.Gain Loss} = \text{RioGrande.Gain Loss} + \text{Storage Adjustment from Elev Vol Table Change}$$

Calculate the Physical Model Net Loss and the Net Loss Adjustment

---

Note: *These values are used for reporting only. They are not used in later calculations.*

$$precip = \text{Precipitation Rate} \times Timestep \times \text{Surface Area}$$

$$\text{Physical Model Net Loss} = \text{Evaporation} - precip$$

A positive value for Physical Model Net Loss represents a loss.

$$totalSanJuanGainLoss = \sum sanJuanGainLosses + \sum sanJuanPassThruGainLosses$$

A negative value for *totalSanJuanGainLoss* represents a loss.

$$totalRioGrandeLossTemp = hypotheticalLoss - preResLoss$$

A positive value for *totalRioGrandeLossTemp* represents a loss.

$$\text{Net Loss Adjustment} = \text{Physical Model Net Loss} + totalSanJuanGainLoss - totalRioGrandeLossTemp$$

Set summary values in table series slots

The Elephant Butte and El Vado Gain Loss sets the following table series slots. Each table below represents a table series slot. The row names in the following tables represent the column headers in the table series slots.

Preliminary Data

Column	Value
Elevation	*Reservoir*.Pool Elevation
Content	*Reservoir*.Storage
Area	*Reservoir*.Surface Area
Evaporation	*Reservoir*.Evaporation
Precipitation	*Reservoir*.Precipitation Volume
Unidentified Loss	$unidentifiedLoss = Max(0, -(\text{Storage} - \text{Storage(t-1)} + \text{Outflow} \times Timestep + \text{Evaporation} - \text{Precipitation Volume} - \text{Storage Adjustment from Elev Vol Table Change}))$
Outflow (cfs)	*Reservoir*.Outflow
Outflow (af/d)	*Reservoir*.Outflow
Inflow (cfs)	*Reservoir*.Evaporation
Inflow (af/d)	*Reservoir*.Evaporation

PreReservoir Condition

Column	Value
River Channel Area	Control and PreRes Areas(Pre-Res River Area)
Barren Area	Control and PreRes Areas(Pre-Res Barren Area)
Meadow and Town Area	Control and PreRes Areas(Pre-Res Meadow Area)
River Area Loss	$preResRiverLoss = preResRiverArea \times evapDepth$
Barren Area Loss	$preResBarrenLoss = preResBarrenArea \times effectPrecipDepth$
Irrigated Area Loss	$preResIrrigLoss = preResIrrigArea \times \text{PreRes Irrigated Area Loss Rate} \times (1 - \text{Surface Ice Coverage}) \times Timestep$
Meadow and Town Area Loss	$preResMeadowLoss = preResMeadowArea \times \text{PreRes Meadow Area Loss Rate} \times (1 - \text{Surface Ice Coverage}) \times Timestep$
PreRes Condition Loss	$preResLoss = preResBarrenLoss + preResIrrigLoss + preResMeadowLoss + preResRiverLoss$

Present Condition

Column	Value
Water Surface Elev	Pool Elevation
Lake Surface Area	Surface Area
River Channel Area	presRiverArea
Barren Area	presBarrenArea
Total Area	Control and PreRes Areas(Control Area)
Water Surface Area Loss	$presWaterSurfaceLoss = (\text{Surface Area} + presRiverArea) \times evapDepth$
Barren Area Loss	$presBarrenLoss = presBarrenArea \times effectPrecipDepth$
Present Condition Loss	$presentLoss = presBarrenLoss + presWaterSurfaceLoss$
PreRes Condition Loss	$preResLoss = preResBarrenLoss + preResIrrigLoss + preResMeadowLoss + preResRiverLoss$
Net RG Plus SJ-C Loss	$presentLoss - preResLoss$

Hypothetical Condition

Column	Value
Total Content	*Reservoir*.Storage
SJ-C Content	totalSanJuanStorageSum
RG Content	totalRioGrandeStorage
Lake Surface Area	hypotheticalSurfaceArea
River Channel Area	hypotheticalRiverArea
Barren Area	hypotheticalBarrenArea
Water Surface Area Loss	$hypotheticalWaterSurfaceLoss = (hypotheticalSurfaceArea + hypotheticalRiverArea) \times evapDepth$
Barren Area Loss	$hypotheticalBarrenLoss = hypotheticalBarrenArea \times effectivePrecipDepth$
Hypo. Condition Loss	$hypotheticalLoss = hypotheticalBarrenLoss + hypotheticalWaterSurfaceLoss$
PreRes Condition Loss	$preResLoss = preResBarrenLoss + preResIrrigLoss + preResMeadowLoss + preResRiverLoss$

Column	Value
Net RG Condition Loss	totalRioGrandeLossTemp
Net RG Plus SJ-C Loss	$presentLoss - preResLoss$
Net SJ-C Loss	$\sum -Account.\mathrm{Gain\ Loss}$, for all accounts with SanJuan water type

Daily Reservoir Computations

Column	Value
Water Surface Elev	*Reservoir*.Pool Elevation
Total Content	*Reservoir*.Storage
Change in Content	$\mathrm{Storage} - \mathrm{Storage(t\text{-}1)}$
RG Release (cfs)	$\sum Account.\mathrm{Outflow}$, for all accounts with RioGrande water type
RG Release (af/d)	$\sum Account.\mathrm{Outflow}$, for all accounts with RioGrande water type
SJ-C Release (cfs)	$\sum Account.\mathrm{Outflow}$, for all accounts with SanJuan water type

Column	Value
SJ-C Release (af/d)	$\sum Account.\text{Outflow}$, for all accounts with SanJuan water type
Evap	*Reservoir*.Evaporation
Precip	*Reservoir*.Precipitation Volume
RG Inflow (cfs)	$\sum Account.\text{Inflow}$, for all accounts with RioGrande water type
RG Inflow (af/d)	$\sum Account.\text{Inflow}$, for all accounts with RioGrande water type
SJ-C Inflow (cfs)	$\sum Account.\text{Inflow}$, for all accounts with SanJuan water type
SJ-C Inflow (af/d)	$\sum Account.\text{Inflow}$, for all accounts with SanJuan water type
Total Inflow (cfs)	*Reservoir*.Inflow
Total Inflow (af/d)	*Reservoir*.Inflow

Reservoir Summary

Column	Value
Total Drawdown	Storage(t-1) – Storage – Storage Adjustment from Elev Vol Table Change

Column	Value
RG Loss	totalRioGrandeLossTemp
SJ-C Loss	$\sum -Account.\text{Gain Loss}$, for all accounts with SanJuan water type
Total Loss	SJ-C Loss + totalRioGrandeLossTemp
RG Release From Storage	$rioGrandePrevStorSum - totalRioGrandeStorage - totalRioGrandeLossTemp$
SJC Release From Storage	$totalPrevSanJuanStorage - totalSanJuanStorageSum + \sum AllSanJuanAccounts.\text{Gain Loss}$
RG Content	totalRioGrandeStorage
SJ-C Content	totalSanJuanStorageSum
Total Content	*Reservoir*.Storage

Nambe Falls Gain Loss

This method calculates the total San Juan losses as the difference between the Present Condition Loss and the PreReservoir Condition Loss. The total San Juan loss is then distributed to all San Juan accounts proportional to the storage in each account at the end of the previous timestep. This allows the San Juan Storages to solve at the current timestep.

$$totalSanJuanLosses = -(presentLoss - preResLoss)$$

$$\boldsymbol{Account}\textbf{.Gain Loss} = \frac{Account.\text{Storage(t-1)}}{totalPrevSanJuanStorage} \times totalSanJuanLosses$$

The method also calculates the Net Loss Adjustment, Operational Release and Irrigation Release.This method does not set the Gain Loss for the Rio Grande account(s).

Details of the individual components are provided in "Slots specific to the Nambe Falls Gain Loss method." The steps carried out by this method are detailed in "Steps carried out by the Nambe Falls Gain Loss method" on page 247.

Slots specific to the Nambe Falls Gain Loss method

Selecting the Nambe Falls Gain Loss method instantiates the following slots:

+ Augmented Release Switch

Type: Series Slot
Units: None
Description: Specifies whether drawdown should go toward Operational Release or Irrigation Release
The value in this slot must be 0 or 1. If the value is 1, any drawdown is counted as Operational Release. If the value is 0, any drawdown is counted as Irrigation Release.
I/O: Required Input
Supply Links: No

+ Control and PreRes Areas

Type: Table Slot
Units: Area, Area, Area, Area, Area, Area
Description: The table contains a one row and six columns, one for each of the following: Control Area, Pre-Res Barren Area, Pre-Res Irrigated Area, Pre-Res Meadow Area, Pre-Res River Area, Pre-Res Lake Area
The areas in this table are used for calculating losses for pre-reservoir conditions.
I/O: Required Input
Supply Links: No

+ Effective Precip

Type: Series Slot
Units: Volume
Description: The effective precipitation calculated for the given timestep.
The total accumulated precipitation depth for the month is used to calculate the accumulated effective precipitation for the current day and the previous day. The difference is the Effective Precip for the current day. For a monthly timestep, the effective precipitation is calculated from the total precipitation for the current timestep.
I/O: Output
Supply Links: No

+ Effective Precipitation Table

Type: Table Slot

Units: Length, Fraction, Length

Description: The relationship between total accumulated precipitation depth for the month and the amount considered effective precipitation.

The first column gives a depth of monthly precipitation. The second column gives the fraction of that precipitation depth that is considered effective precipitation. The third column contains the accumulated effective precipitation value corresponding to the precipitation in the first column. Linear interpolation is used to calculate the effective precipitation depth based on the total precipitation depth for the month. Only the first and third columns are actually used in the interpolation. The second column is only for reference.

I/O: Required Input

Supply Links: No

+ Hypothetical Lake Elev River Area Table

Type: Table Slot

Units: Length, Area

Description: Relationship of Hypothetical River Area to Hypothetical Lake Elevation

This table is used to calculate river area for hypothetical conditions

I/O: Required Input

Supply Links: No

+ Net Loss Adjustment

Type: Series Slot

Units: Volume

Description: The difference between the Physical Model Net Loss and the losses calculated by the accounting method

If this value is positive the physical model loss was greater than the accounting method loss

I/O: Output

Supply Links: No

+ Physical Model Net Loss

Type: Series Slot

Units: Volume

Description: The net loss from the simulation methods (Evaporation - Precipitation).

A positive value represents a net loss

I/O: Output

Supply Links: No

+ PreRes Irrigated Area Loss Rate

Type: Series Slot with Periodic Input

Units: Velocity
Description: The loss rate over irrigated area before the reservoir existed
This value is used for calculating losses for pre-reservoir conditions.
I/O: Required Input
Supply Links: No

PreRes Meadow Area Loss Rate

Type: Series Slot with Periodic Input
Units: Velocity
Description: The loss rate over meadow area before the reservoir existed
This value is used for calculating losses for pre-reservoir conditions.
I/O: Required Input
Supply Links: No

Present Lake Elev River Area Table

Type: Table Slot
Units: Length, Area
Description: Relationship of River Area to Pool Elevation
This table is used to calculate river area for present conditions
I/O: Required Input
Supply Links: No

The following table series slots are also set by this method. Details of each column value are given at the end of the method description. See

- “Preliminary Data” on page 252.
- “PreReservoir Condition” on page 253
- “Present Condition” on page 254
- “Nambe Daily Reservoir Comps” on page 255

Steps carried out by the Nambe Falls Gain Loss method

The Nambe Falls Gain Loss method carries out the following steps:

1. Check for previous Storage and Accrual on all accounts.
2. Get lists of each type of San Juan account slot.
3. Calculate Effective Precipitation. See “Calculate effective precipitation” on page 248.
4. Calculate the evaporation depth and effective precipitation depth to use in loss calculations. See “Calculate the evaporation depth and effective precipitation depth to use in loss calculations” on page 249.

5. Calculate the individual components of the PreReservoir Condition Loss and the total PreRes Condition Loss. See "Calculate the individual components of the PreReservoir Condition Loss and the total PreRes Condition Loss" on page 249.
6. Calculate the individual components of the Present Condition Loss and the total Present Condition Loss. See "Calculate the individual components of the Present Condition Loss and the total Present Condition Loss" on page 250.
7. Distribute the San Juan gains/losses. See "Distribute San Juan gains/losses" on page 250.
8. Calculate the physical model net loss and the net loss adjustment. See "Calculate the Physical Model Net Loss and the Net Loss Adjustment" on page 251.
9. Calculate the operational release, irrigation release, and storage replacement. See "Calculate the Operational Release, Irrigation Release and Storage Replacement" on page 251.
10. Set the summary values in the table series slots. See "Set summary values in table series slots" on page 252.

Calculate effective precipitation

For a monthly timestep, this can be done based on the Precipitation Volume for the month, which will have been set in simulation. The Precipitation Volume calculation will already account for Ice Coverage if applicable. For a daily timestep, Effective Precipitation must be calculated using the cumulative sum of Precipitation Rate from the beginning of the month to the current day, and Ice Coverage is included explicitly. At a monthly timestep the average surface area is used, whereas the current (end of timestep) surface area is used for a daily timestep.

IF isMonthly

$$precipRate = \frac{\text{Precipitation Volume}}{(\text{Surface Area} + \text{Surface Area(t-1)})/2}$$

$$effectPrecipRate = TableInterpolation(\text{Effective Precipitaiton Table}, precipRate)$$

$$\text{Effective Precip} = effectPrecipRate \times \frac{\text{Surface Area} + \text{Surface Area(t-1)}}{2}$$

ELSE

$$prevCumulPrecip = \sum_{d = BegMonth}^{CurDay - 1} (\text{Precipitation Rate(d)} \times Timestep$$

$$\times (1 - \text{Surface Ice Coverage(d)}))$$

$$cumulPrecip = \sum_{d = BegMonth}^{CurDay} (\text{Precipitation Rate(d)} \times Timestep$$

$$\times (1 - \text{Surface Ice Coverage(d)}))$$

$$prevEffectPrecip = TableInterpolation(\text{Effective Precipitaiton Table}, prevCumulPrecip)$$

$$presentEffectPrecip = TableInterpolation(\text{Effective Precipitaiton Table}, cumulPrecip)$$

$$effectPrecipRate = presentEffectPrecip - prevEffectPrecip$$

$$\text{Effective Precip} = effectPrecipRate \times \text{Surface Area}$$

END IF

Calculate the evaporation depth and effective precipitation depth to use in loss calculations

For all following calculations, Area values for a daily timestep are current (end of timestep) areas. For a monthly timestep, Area values are averages of the end of the previous timestep and end of the current timestep values, similarly for Ice Coverage. If Ice Coverage is not applicable based on the selected evaporation method, it is zero in the calculations below.

$$evapDepth = \frac{\text{Evaporation}}{\text{Surface Area}}$$

$$effectivePrecipDepth = \frac{\text{Effective Precip}}{\text{Surface Area} \times (1 - \text{Surface Ice Coverage})}$$

If the Surface Area is zero, then *evapDepth* and *effecitvePrecipDepth* are zero, and if Ice Coverage is one, then *effectivePrecipDepth* is zero.

Calculate the individual components of the PreReservoir Condition Loss and the total PreRes Condition Loss

Each of these values is positive for a loss. The PreRes areas are contained in the Control and PreRes Areas table slot. The results of these calculations are set in the PreReservoir Condition table series slot.

$$preResBarrenLoss = preResBarrenArea \times effectPrecipDepth$$

$$preResIrrigLoss = preResIrrigArea \times \text{PreRes Irrigated Area Loss Rate} \times (1 - \text{Surface Ice Coverage}) \times Timestep$$

$$preResMeadowLoss = preResMeadowArea \times \text{PreRes Meadow Area Loss Rate} \times (1 - \text{Surface Ice Coverage}) \times Timestep$$

$$preResRiverLoss = preResRiverArea \times evapDepth$$

$$preResLoss =$$
$$preResBarrenLoss + preResIrrigLoss + preResMeadowLoss + preResRiverLoss$$

Calculate the individual components of the Present Condition Loss and the total Present Condition Loss

Each of these values is positive for a loss. The results of these calculations are set in the Present Condition table series slot.

$$presRiverArea = TableInterpolation(\text{Present Lake Elev River Area Table, Pool Elevation})$$

$$presBarrenArea = totalArea - presRiverArea - \text{Surface Area}$$

$$presBarrenLoss = presBarrenArea \times effectPrecipDepth$$

$$presWaterSurfaceLoss = (\text{Surface Area} + presRiverArea) \times evapDepth$$

$$netInflowVolume = \text{Storage} - \text{Storage(t-1)} + \text{Outflow} \times Timestep$$
$$+ \text{Evaporation} - \text{Precipitation Volume} - \text{Storage Adjustment from Elev Vol Table Change}$$

The *netInflowVolume* will be negative if there is unidentified loss. If *netInflowVolume* is positive, then unidentified loss is zero. The Storage Adjustment from Elev Vol Table Change component is only included if the Time Varying Elevation Volume method is selected for the Sediment category.

$$unidentifiedLoss = Max(0, -netInflowVolume)$$

unidentifiedLoss will be positive for a loss.

$$presentLoss = presBarrenLoss + presWaterSurfaceLoss + unidentifiedLoss$$

Distribute San Juan gains/losses

The total Gain/Loss allocated to San Juan storage accounts is the difference between the Present Condition Loss and the PreReservoir Condition Loss. A positive value for each of these components represents a loss. In the Gain Loss slot on each account, a positive value represents a gain, and a negative value represents a loss. Thus the negative of the difference is taken before distributing the loss to the Gain Loss slot. The loss is distributed based on the contributing percentage of the total storage at the end of the previous timestep. If the total San Juan storage at the end of the previous timestep was zero, then the Gain Loss is zero for all accounts.

$$totalSanJuanLosses = -(presentLoss - preResLoss)$$

$$sanJuanPrevStorSum = \sum sanJuanStorages(t-1)$$

$$sanJuanPsThruPrevStorSum = \sum sanJuanPassThruStorages(t-1)$$

sanJuanStorages and *sanJuanPassThruStorages* are lists of storage slots on all accounts with the San Juan water type.

$$totalPrevSanJuanStorage = sanJuanPrevStorSum + sanJuanPsThruPrevStorSum$$

$$Account\text{.Gain Loss} = \frac{Account\text{.Storage(t-1)}}{totalPrevSanJuanStorage} \times totalSanJuanLosses$$

This will allow the San Juan accounts to solve for storage.

Calculate the Physical Model Net Loss and the Net Loss Adjustment

$$precip = \text{Precipitation Rate} \times Timestep \times \text{Surface Area}$$

$$\text{Physical Model Net Loss} = \text{Evaporation} - precip$$

A positive value for Physical Model Net Loss represents a loss. A negative value for *totalSanJuanLosses* represents a loss.

$$\text{Net Loss Adjustment} = \text{Physical Model Net Loss} + totalSanJuanLosses$$

Calculate the Operational Release, Irrigation Release and Storage Replacement

If the reservoir elevation was increasing or if the reservoir was spilling, the change in storage is counted as *risingChangeStorage* otherwise it is *droppingChangeStorage*. Note that for these calculations, if the reservoir changes conditions within the month (rising to dropping or dropping to rising) the results at a monthly timestep will not necessarily match the monthly aggregate of results from a daily timestep.

IF $(\text{Pool Elevation} > unregulatedSpillwayCrest)$ OR $(\text{Pool Elevation} \geq \text{Pool Elevation(t-1)})$

$$replacementLoss = -totalSanJuanLosses$$

$$risingChangeStorage = \text{Storage} - \text{Storage(t-1)}$$

$$droppingChangeStorage = 0$$

ELSE

$$replacementLoss = 0$$

$$risingChangeStorage = 0$$

$$droppingChangeStorage = \text{Storage} - \text{Storage(t-1)}$$

END IF

If the reservoir was not spilling at the end of the timestep but was spilling at the start, then these values need to be modified.

IF $(\text{Pool Elevation} < unregulatedSpillwayCrest)$ AND
$(\text{Pool Elevation(t-1)} > unregulatedSpillwayCrest)$

$$storageAtSpillway = TableInterpolation(\text{Elevation Volume Table}, unregulatedSpillwayCrest)$$

$$droppingChangeStorage = storageAtSpillway - \text{Storage}$$

$$risingChangeStorage = \text{Storage} - \text{Storage(t-1)} + droppingChangeStorage$$

END IF

$$storageReplacement = risingChangeStorage$$

$$totalReplacement = replacementLoss + storageReplacement$$

IF Augmented Release Switch = 1

$$operationgalRelease = droppingChangeStorage$$

$$irrigationRelease = 0$$

ELSE

$$operationgalRelease = 0$$

$$irrigationRelease = droppingChangeStorage$$

END IF

Set summary values in table series slots

The Nambe Falls Gain/Loss method sets the following table series slots. Each table below represents a table series slot. The row names in the following tables represent the column headers in the table series slots.

Preliminary Data

Column	Value
Elevation	*Reservoir*.Pool Elevation
Content	*Reservoir*.Storage
Area	*Reservoir*.Surface Area
Evaporation	*Reservoir*.Evaporation
Precipitation	*Reservoir*.Precipitation Volume

Column	Value
Unidentified Loss	$unidentifiedLoss = Max(0, -(\text{Storage} - \text{Storage(t-1)} + \text{Outflow} \times Timestep + \text{Evaporation} - \text{Precipitation Volume} - \text{Storage Adjustment from Elev Vol Table Change}))$
Outflow (cfs)	*Reservoir*.Outflow
Outflow (af/d)	*Reservoir*.Outflow
Inflow (cfs)	*Reservoir*.Evaporation
Inflow (af/d)	*Reservoir*.Evaporation

PreReservoir Condition

Column	Value
River Channel Area	Control and PreRes Areas(Pre-Res River Area)
Barren Area	Control and PreRes Areas(Pre-Res Barren Area)
Meadow and Town Area	Control and PreRes Areas(Pre-Res Meadow Area)
River Area Loss	$preResRiverLoss = preResRiverArea \times evapDepth$

Column	Value
Barren Area Loss	$preResBarrenLoss = preResBarrenArea \times effectPrecipDepth$
Irrigated Area Loss	$preResIrrigLoss = preResIrrigArea \times \text{PreRes Irrigated Area Loss Rate} \times (1 - \text{Surface Ice Coverage}) \times Timestep$
Meadow and Town Area Loss	$preResMeadowLoss = preResMeadowArea \times \text{PreRes Meadow Area Loss Rate} \times (1 - \text{Surface Ice Coverage}) \times Timestep$
PreRes Condition Loss	$preResLoss = preResBarrenLoss + preResIrrigLoss + preResMeadowLoss + preResRiverLoss$

Present Condition

Column	Value
Water Surface Elev	*Reservoir*.Pool Elevation
Lake Surface Area	*Reservoir*.Surface Area
River Channel Area	presRiverArea
Barren Area	presBarrenArea
Total Area	Control and PreRes Areas(Control Area)

Column	Value
Water Surface Area Loss	$presWaterSurfaceLoss = (\text{Surface Area} + presRiverArea) \times evapDepth$
Barren Area Loss	$presBarrenLoss = presBarrenArea \times effectPrecipDepth$
Present Condition Loss	$presentLoss = presBarrenLoss + presWaterSurfaceLoss$
PreRes Condition Loss	$preResLoss = preResBarrenLoss + preResIrrigLoss + preResMeadowLoss + preResRiverLoss$
Net RG Plus SJ-C Loss	totalSanJuanLosses

Nambe Daily Reservoir Comps

Column	Value
Water Surface Elev	*Reservoir*.Pool Elevation
Total Content	*Reservoir*.Storage
Change in Content	$Reservoir.\text{Storage} - Reservoir.\text{Storage(t-1)}$

Column	Value
Storage Replacement	storageReplacemet
Total Release (cfs)	*Reservoir*.Outflow
Total Release (af/d)	*Reservoir*.Outflow
Operational Release	operationalRelease
Irrigation Release	irrigationRelease
Net Loss	totalSanJuanLosses
Replacement Loss	replacementLoss
Total Inflow (cfs)	$\left(\frac{Reservoir.\text{Storage} - Reservoir.\text{Storage(t-1)} + totalSanJuanLosses}{Timestep}\right) +$ $Reservoir.\text{Outflow}$
Total Inflow (af/d)	$\left(\frac{Reservoir.\text{Storage} - Reservoir.\text{Storage(t-1)} + totalSanJuanLosses}{Timestep}\right) +$ $Reservoir.\text{Outflow}$
Storage + Loss Replmt.	$totalReplacement = replacementLoss + storageReplacement$

Heron Gain Loss

This method calculates the Gain Loss for accounts on the Heron Reservoir. The method is currently configured to function correctly for daily and monthly timesteps only.This method requires simulation to have solved for Storage, Evaporation and Precipitation Volume, so the execution time must be set to After Simulation.

The Gain Loss for the RioGrande account is set equal to seepage.

$$\text{RioGrande.Gain Loss} = -\text{Seepage} \times \mathit{Timestep}$$

This allows the RioGrande account to solve for storage. All other gains/losses go to the FederalSanJuan account. The Gain Loss for all other accounts with the SanJuan water type is set to zero.

$$sanJuanPrevStorSum = \sum sanJuanStorages(t-1)$$

$$sanJuanOutflowVol = \sum sanJuanOutflows \times Timestep$$

sanJuanStorages and *sanJuanOutflows* lists of the Outflow and Storage slots on all accounts withe the San Juan water type.

$$\text{FederalSanJuan.Gain Loss} = \mathit{Reservoir}.\text{Storage} - \text{RioGrande.Storage} - sanJuanPrevStorSum - \text{FederalSanJuan.Inflow} \times Timestep + sanJuanOutflowVol$$

$\mathit{Account}.\text{Gain Loss} = 0$, for all other San Juan accounts.

Details of the individual components are provided in "Slots specific to the Heron Gain Loss method." The table series slots that are set by the Heron Gain Loss method are detailed in “Summary values in table series slots set by the Heron Gail Loss method” on page 258.

Slots specific to the Heron Gain Loss method

Selecting the Heron Gain Loss method instantiates the following slots:

+ Net Loss Adjustment

Type: Series Slot
Units: Volume
Description: The difference between the Physical Model Net Loss and the losses calculated by the accounting method
If this value is positive the physical model loss was greater than the accounting method loss
I/O: Output
Supply Links: No

+ Physical Model Net Loss

Type: Series Slot
Units: Volume
Description: The net loss from the simulation methods (Evaporation - Precipitation).
A positive value represents a net loss

I/O: Output
Supply Links: No

$$\text{Net Loss Adjustment} = \text{Physical Model Net Loss} + \text{FederalSanJuan.Gain Loss}$$

$$\text{Physical Model Net Loss} = \text{Evaporaton} - \text{Precipitation Volume}$$

Physical Model Net Loss is positive for a loss. The account Gain Loss slot values are negative for a loss.

Summary values in table series slots set by the Heron Gail Loss method

The Heron Gain Loss method sets the following table series slots. Each table below represents a table series slot. The row names in the following tables represent the column headers in the table series slots.

Preliminary Data

Column	Value
Elevation	*Reservoir*.Pool Elevation
Content	*Reservoir*.Storage
Area	*Reservoir*.Surface Area
Evaporation	*Reservoir*.Evaporation
Precipitation	*Reservoir*.Precipitation Volume
Unidentified Loss	$unidentifiedLoss = Max(0, -(\text{Storage} - \text{Storage(t-1)} + \text{Outflow} \times Timestep + \text{Evaporation} - \text{Precipitation Volume} - \text{Storage Adjustment from Elev Vol Table Change}))$
Outflow (cfs)	*Reservoir*.Outflow

Column	Value
Outflow (af/d)	*Reservoir*.Outflow
Inflow (cfs)	*Reservoir*.Evaporation
Inflow (af/d)	*Reservoir*.Evaporation

Daily Reservoir Computations

Column	Value
Water Surface Elev	*Reservoir*.Pool Elevation
Total Content	*Reservoir*.Storage
Change in Content	$\text{Storage} - \text{Storage(t-1)}$
RG Release (cfs)	$\sum Account.\text{Outflow}$, for all accounts with RioGrande water type
RG Release (af/d)	$\sum Account.\text{Outflow}$, for all accounts with RioGrande water type
SJ-C Release (cfs)	$\sum Account.\text{Outflow}$, for all accounts with SanJuan water type
SJ-C Release (af/d)	$\sum Account.\text{Outflow}$, for all accounts with SanJuan water type

Column	Value
Evap	*Reservoir*.Evaporation
Precip	*Reservoir*.Precipitation Volume
RG Inflow (cfs)	$\sum Account.\text{Inflow}$, for all accounts with RioGrande water type
RG Inflow (af/d)	$\sum Account.\text{Inflow}$, for all accounts with RioGrande water type
SJ-C Inflow (cfs)	$\sum Account.\text{Inflow}$, for all accounts with SanJuan water type
SJ-C Inflow (af/d)	$\sum Account.\text{Inflow}$, for all accounts with SanJuan water type
Total Inflow (cfs)	*Reservoir*.Inflow
Total Inflow (af/d)	*Reservoir*.Inflow

Reservoir Summary

Column	Value
Total Drawdown	Storage(t-1) – Storage – Storage Adjustment from Elev Vol Table Change
RG Loss	0

Column	Value
SJ-C Loss	$\sum -Account.\text{Gain Loss}$, for all accounts with SanJuan water type
Total Loss	SJ-C Loss
RG Release From Storage	$\text{RioGrande.Storage(t-1)} - \text{RioGrande.Storage}$
SJC Release From Storage	$totalPrevSanJuanStorage - totalSanJuanStorageSum + \sum AllSanJuanAccounts.\text{Gain Loss}$
RG Content	RioGrande.Storage
SJ-C Content	totalSanJuanStorageSum
Total Content	*Reservoir*.Storage

For a daily timestep, values are set in three additional table series slots:

- Preliminary Data Totals
- Daily Reservoir Totals
- Reservoir Summary Totals

These slots contain the monthly totals of their corresponding table series slots. The monthly sums are set at the last timestep for each month. All other cells display “NaN.” For a monthly timestep, no values will be set in these slots.

Reservoir Account Slot Inflow

The reservoir has a single basin specific method. This is described in "Heron Inflow."

Heron Inflow

This method sets the Slot Inflow for all accounts on Heron Reservoir. The Heron Inflow method requires that a method be selected in the Seepage category. The method requires values calculated by simulation, such as seepage and surface area; therefore the execution time must be set to After Simulation. This method is currently configured to function correctly for daily and monthly timesteps only.

The Slot Inflow for all accounts with San Juan water type is set to zero. The Slot Inflow for the RioGrande account is set to the difference between the change in cumulative natural flow and the Inflow to the Rio Grande account.

$Account.\text{Slot Inflow} = 0$, for all SanJuan accounts

$$\text{RioGrande.Slot Inflow} = \frac{naturalFlowCumul - prevNaturalFlowCumul}{Timestep} - \text{RioGrande.Inflow}$$

The calculation of *naturalFlowCumul* is dependent on whether the RioGrande.Inflow is in high flow or low flow conditions for the month. For a daily timestep, if the condition changes at some point during the month, all values are recalculated previous timesteps in the month based on the new condition. For a monthly timestep, the monthly condition is known at the end of each timestep (end of each month) so there is no need to recalculate any values.

The details of the individual components are provided in "Slots specific to the Heron Inflow method." The steps carried out by the Heron Inflow method are detailed in "Steps carried out by the Heron Inflow method" on page 263.

Slots specific to the Heron Inflow method

Selecting the Heron Inflow method instantiates the following slots:

+ Heron Inflow Coefficients

Type: Table Slot
Units: Volume, None, None
Description: The first column contains the test volume for determining whether the total monthly RioGrande flow is high flow or low flow. The second and third columns contain the ratios used for low flow and high flow conditions.
The products of the flow ratios and the RioGrande.Inflow are used in calculating the Natural Flow.
I/O: Required Input
Supply Links: No

+ Rio Grande Ratio

Type: Series Slot
Units: None

Description:	The final flow ratio used at each timestep based on the monthly flow condition The value in this slot will be either the Low Flow Ratio or the High Flow Ratio from the Heron Inflow Coefficients table slot.
I/O:	Output only
Supply Links:	No

Effective Precip

Type:	Series Slot
Units:	Volume
Description:	The effective precipitation calculated for the given timestep. The total accumulated precipitation depth for the month is used to calculate the accumulated effective precipitation for the current day and the previous day. The difference is the Effective Precip for the current day. For a monthly timestep, the effective precipitation is calculated from the total precipitation for the current timestep.
I/O:	Output
Supply Links:	No

Effective Precipitation Table

Type:	Table Slot
Units:	Length, Fraction, Length
Description:	The relationship between total accumulated precipitation depth for the month and the amount considered effective precipitation. The first column gives a depth of monthly precipitation. The second column gives the fraction of that precipitation depth that is considered effective precipitation. The third column contains the accumulated effective precipitation value corresponding to the precipitation in the first column. Linear interpolation is used to calculate the effective precipitation depth based on the total precipitation depth for the month. Only the first and third columns are actually used in the interpolation. The second column is only for reference.
I/O:	Required Input
Supply Links:	No

The following table series slots are also set by this method. Details of each column value are given in the method description below. See:

- "Heron Inflow Values" on page 266.
- "Heron Cumulative Inflow Values" on page 270

Steps carried out by the Heron Inflow method

The Heron Inflow method carries out the following steps:

1. Set the slot inflow for all accounts with San Juan water type to zero. See "Set the Slot Inflow for all accounts with San Juan water type to zero" on page 264.

2. Calculate the effective precipitation. See "Calculate Effective Precipitation" on page 264.
3. Calculate the test volume and the cumulative RioGrande.Inflow for the month to date. See "Calculate the test volume and the cumulative RioGrande.Inflow for the month to date" on page 265.
4. Convert storages to equivalent flows based on the timestep length for use in net gain calculations. See "Convert storages to equivalent flows based on the timestep length for use in net gain calculations" on page 266.
5. Calculate local variables and set the vlaues in the Heron Inflow Values table series slot. See "Calculate local variables and set the values in the Heron Inflow Values table series slot" on page 266.
6. Check flow conditions and then set the inflow ratio and monthly cumulative values accordingly. See "Check the flow conditions, and then set the inflow ratio and monthly cumulative values accordingly" on page 267.
7. Calculate the ratio inflow and the cumulative value net gain. See "Calculate the cumulative ratio inflow and the cumulative net gain" on page 269.
8. Determine the correct logic to use. See "Determine the correct logic to use" on page 270.
9. Set the values in the Heron Cumulative Inflow Vlaues table series slot. See "Set the values in the Heron Cumulative Inflow Values table series slot" on page 270.
10. Set the RioGrande.Slot inflow. See "Set RioGrande.Slot Inflow" on page 271.
11. Modify the previous logic. See "Set the previous logic to the current logic to use for the next timestep" on page 273.
12. Calculate the monthly sum of each column in the Heron Inflow Values table series slot. See "Calculate the monthly sum of each column in the Heron Inflow Values table series slot" on page 273.
13. Set the inflowRatioCumul and adjust the netGainCumul. See "Set inflowRatioCumul and adjust netGainCumul if inflowRatioMonthly is negative" on page 273.
14. Set the cumulative natural flow. See "Set the cumulative natural flow based on the current (controller timestep) logic" on page 274.
15. Set the values in the Heron Cumulative Inflow Values table series slot. See "Set the values in the Heron Cumulative Inflow Values table series slot at timestep d" on page 275.
16. Set the RioGrande.Slot Inflow at timestep d. See "Set the RioGrande.Slot Inflow at timestep d to difference between the natural flow at this timestep and the RioGrande.Inflow at this timestep" on page 275.

Set the Slot Inflow for all accounts with San Juan water type to zero

$Account$.Slot Inflow $= 0$, for all San Juan accounts

Calculate Effective Precipitation

For a monthly timestep, this can be done based on the Precipitation Volume for the month, which will have been set in simulation. The Precipitation Volume calculation will already account for Ice Coverage if applicable. For a daily timestep, Effective Precipitation must be calculated using the cumulative sum of Precipitation Rate from the beginning of the month to the current day, and Ice Coverage is included explicitly. At a monthly timestep the average surface area is used, whereas the current (end of timestep) surface area is used for a daily timestep.

IF isMonthly

$$precipRate = \frac{\text{Precipitation Volume}}{(\text{Surface Area} + \text{Surface Area(t-1)})/2}$$

$$effectPrecipRate = TableInterpolation(\text{Effective Precipitaiton Table}, precipRate)$$

$$\text{Effective Precip} = effectPrecipRate \times \frac{\text{Surface Area} + \text{Surface Area(t-1)}}{2}$$

ELSE

$$prevCumulPrecip = \sum_{d = BegMonth}^{CurDay - 1} (\text{Precipitation Rate(d)} \times Timestep \times (1 - \text{Ice Coverage(d)}))$$

$$cumulPrecip = \sum_{d = BegMonth}^{CurDay} (\text{Precipitation Rate(d)} \times Timestep \times (1 - \text{Ice Coverage(d)}))$$

$$prevEffectPrecip = TableInterpolation(\text{Effective Precipitaiton Table}, prevCumulPrecip)$$

$$presentEffectPrecip = TableInterpolation(\text{Effective Precipitaiton Table}, cumulPrecip)$$

$$effectPrecipRate = presentEffectPrecip - prevEffectPrecip$$

$$\text{Effective Precip} = effectPrecipRate \times \text{Surface Area}$$

END IF

Calculate the test volume and the cumulative RioGrande.Inflow for the month to date

For a daily timestep, these are cumulative values from the beginning of the month to the current timestep. For a monthly timestep, the cumulative monthly value is simply the current timestep value.

IF isMonthly

$$testVolume = \text{Heron Inflow Coefficients(Volume to Test)}$$

$$rgInflowMonthlyVol = \text{RioGrande.Inflow} \times Timestep$$

ELSE (daily timestep)

IF $dayOfMonth < 30$

$$testVolume = \frac{\text{Heron Inflow Coefficients(Volume to Test)}}{30} \times dayOfMonth$$

ELSE

$$testVolume = \text{Heron Inflow Coefficients(Volume to Test)}$$

END IF

$$rgInflowMonthlyVol = \sum_{d = BegMonth}^{CurDay} (\text{RioGrande.Inflow(d)} \times Timestep)$$

END IF

Convert storages to equivalent flows based on the timestep length for use in net gain calculations

$$storageAsFlow = \frac{Reservoir\text{.Storage}}{Timestep}$$

$$prevStorageAsFlow = \frac{Reservoir\text{.Storage(t-1)}}{Timestep}$$

Calculate local variables and set the values in the Heron Inflow Values table series slot

The values that you must calculate are displayed under Column and you set the values in the Heron Inflow Values table series slot.

Heron Inflow Values

Column	Value
Effective Precip (Volume)	*Reservoir*.Effective Precip
Net Evaporation (Volume)	*Reservoir*.Evaporation - *Reservoir*.Effective Precip
Seepage (Flow)	*Reservoir*.Seepage
Low Ratio (Flow)	$inflowRatioLow = \text{RioGrande.Inflow} \times \text{Heron Inflow Coefficients(Low Flow Ratio)}$

Column	Value
High Ratio (Flow)	$\boldsymbol{inflowRatioHigh} =$ RioGrande.Inflow $\times$ Heron Inflow Coefficients(High Flow Ratio)
Low Net EOD Gain (Flow)	$\boldsymbol{netGainLow} = storageAsFlow - prevStorageAsFlow + Reservoir.\text{Outflow} + Reservoir.\text{Seepage} - inflowRatioLow - \text{FederalSanJuan.Inflow}$
High Net EOD Gain (Flow)	$\boldsymbol{netGainHigh} = storageAsFlow - prevStorageAsFlow + Reservoir.\text{Outflow} + Reservoir.\text{Seepage} - inflowRatioHigh - \text{FederalSanJuan.Inflow}$
Low Hybrid Seepage (Flow)	$\boldsymbol{hybridSeepageLow} = Reservoir.\text{Seepage} + inflowRatioLow$
High Hybrid Seepage (Flow)	$\boldsymbol{hybridSeepageHigh} = Reservoir.\text{Seepage} + inflowRatioHigh$
Low Hybrid EOD (Flow)	$\boldsymbol{hybridNetGainLow} = netGainLow + inflowRatioLow$
High Hybrid EOD (Flow)	$\boldsymbol{hybridNetGainHigh} = netGainHigh + inflowRatioHigh$
EOD Content (Volume)	N/A (Values are not set in this column)

Check the flow conditions, and then set the inflow ratio and monthly cumulative values accordingly

IF $rgInflowMonthlyVol < testVolume$, (low flow conditions)

$$\text{Rio Grande Ratio} = \text{Heron Inflow Coefficients(Low Flow Ratio)}$$

$inflowRatio = inflowRatioLow$

IF isMonthly

$$inflowRatioMonthly = inflowRatio \times Timestep$$

$$netGainMonthly = netGainLow \times Timestep$$

$$seepageMonthly = \text{Seepage} \times Timestep$$

ELSE (daily timestep)

$$inflowRatioMonthly = \sum_{d = BegMonth}^{CurDay} \text{Heron Inflow Values(Low Ratio, } d) \times Timestep$$

$$netGainMonthly =$$

$$\sum^{CurDay} \text{Heron Inflow Values(Low Net EOD Gain, } d) \times Timestep$$

$$seepageMonthly = \sum_{d = BegMonth}^{CurDay} \text{Seepage} \times Timestep$$

END IF

ELSE (high flow conditions)

$$\text{Rio Grande Ratio} = \text{Heron Inflow Coefficients(High Flow Ratio)}$$

$$inflowRatio = inflowRatioHigh$$

IF isMonthly

$$inflowRatioMonthly = inflowRatio \times Timestep$$

$$netGainMonthly = netGainLow \times Timestep$$

$$seepageMonthly = \text{Seepage} \times Timestep$$

ELSE (daily timestep)

$$inflowRatioMonthly = \sum_{d = BegMonth}^{CurDay} \text{Heron Inflow Values(High Ratio, } d) \times Timestep$$

$$netGainMonthly =$$

$$\sum^{CurDay} \text{Heron Inflow Values}(\text{High Net EOD Gain}, d) \times Timestep$$

$$seepageMonthly = \sum_{d = BegMonth}^{CurDay} \text{Seepage} \times Timestep$$

END IF

END IF

Calculate the cumulative ratio inflow and the cumulative net gain

If it is the first day of the month or a monthly timestep, these are just the *inflowRatio* (if it is positive) and the *netGainMonthly* for the timestep.

IF $(dayOfMonth = 1)$ OR (isMonthly)

IF $inflowRatio \le 0$

$$cumulRatio = 0$$

$$cumulNetGain = netGainMonthly + inflowRatio \times Timestep$$

ELSE (*inflowRatio* is positive)

$$cumulRatio = inflowRatio \times Timestep$$

$$cumulNetGain = netGainMonthly$$

END IF

ELSE (after the first of the month at a daily timestep)

IF $inflowRatioMonthly \le 0$

$$cumulRatio = 0$$

$$cumulNetGain = netGainMonthly + inflowRatioMonthly$$

ELSE

$$prevCumulRatio = \text{Heron Cumulative Inflow Values}(\text{Ratio Inflow}, t - 1)$$

$$cumulRatio = prevCumulRatio + inflowRatio \times Timestep$$

$$cumulNetGain = netGainMonthly$$

END IF

END IF

Determine the correct logic to use

IF $cumulNetGain > seepageMonthly$

$$cumulInflowWithin = cumulNetGain$$

$$currentLogic = NetGain$$

ELSE

$$cumulInflowWithin = seepageMonthly$$

$$currentLogic = Seepage$$

END IF

IF $(cumulNetGain \le 0)$ AND $(cumulRatio > seepageMonthly)$

$$cumulInflowWithin = 0$$

$$currentLogic = Ratio$$

END IF

IF $cumulRatio > seepageMonthly$

$$cumulNaturalFlow = cumulRatio + cumulInflowWithin$$

$$currentLogic = currentLogic + Ratio$$

ELSE

$$cumulNaturalFlow = cumulInflowWithin$$

END IF

Set the values in the Heron Cumulative Inflow Values table series slot

The table values are set to the local variable as follows:

Heron Cumulative Inflow Values

Column Header (Unit Type)	Value (local variable)
Ratio Inflow (Volume)	cumulRatio

Column Header (Unit Type)	Value (local variable)
Net EOD Gain (Volume)	cumulNetGain
Inflow Within (Volume)	cumulInflowWithin
Natural Flow (Volume)	cumulNaturalFlow

Set RioGrande.Slot Inflow

If it is the first day of the month, or a monthly timestep, or if the conditions have not changed from the previous timestep, then the RioGrande.Slot Inflow can be set. If conditions have changed, then a helper function (findHeronRGInflow) is called. This function recalculates values for previous timesteps in the month based on the conditions at the current timestep.

IF $(dayOfMonth = 1)$ OR isMonthly

$$\text{RioGrande.Slot Inflow} = \frac{cumulNaturalFlow}{Timestep} - \text{RioGrande.Inflow}$$

ELSE IF $(\text{Rio Grande Ratio} \neq \text{Rio Grande Ratio(t-1)})$ OR $currentLogic \neq previousTimestepLogic$

The first condition here indicates that the flow condition has changed from high to low or low to high. The second condition indicates that the logic has changed from the previous timestep.

- FOR (*d* IN *BeginningOfMonth* TO *CurrentDay*)
  - Set all previous values in Rio Grande Ratio for the month equal to the value for the current day

$$\text{Rio Grande Ratio}(d) = \text{Rio Grande Ratio}(CurrentDay)$$

  - Call findHeronRGInflow to recalculate values for previous timesteps in the month and to set RioGrande.Slot Inflow for all previous days in the month and the current day.

Note: *See “findHeronRGInflow function” for a description of the findHeronRGInflow function.*

END FOR

ELSE (conditions have not changed from the previous day)

$$prevCumulNaturalFlow = \text{Heron Cumulative Inflow Values}(\text{Natural Flow}, t-1)$$

$$\text{RioGrande.Slot Inflow} =$$

$$\frac{cumulNaturalFlow - prevCumulNaturalFlow}{Timestep} - \text{RioGrande.Inflow}$$

END IF

findHeronRGInflow function

The findHeronRGInflow function is called from within the Heron Inflow method when the conditions at the current timestep (high flow/low flow or the logic used) have changed from the previous timestep. The function recalculates previous values for the given month used in the RioGrande.Slot Inflow calculation based on the conditions at the current timestep. This recalculation only applies for a daily timestep. At a monthly timestep, the final condition for the month is know at the end of each timestep when values ar first calculated.

The steps in the function are described below.

Note that this function is called from within a FOR loop over all timesteps from the first day of the month to the current controller timestep. In the notation below, the index *d* refers to the timestep in the FOR loop, and the index *curDay* refers to the current controller timestep.

IF $\text{Rio Grande Ratio}(curDay) = \text{Heron Inflow Coefficients(Low Flow Ratio)}$

- This would indicate low flow conditions for the month to-date
  - Set the *inflowRatio* for the day index in the FOR loop equal to the low inflow ratio, and sum the cumulative inflow ratio, net gain and seepage from the beginning of the month to index day.

$$inflowRatio(d) = \text{Heron Inflow Values}(\text{Low Ratio}, d)$$

$$inflowRatioNet = \sum_{t = BegMonth}^{d} \text{Heron Inflow Values}(\text{Low Ratio}, t) \times Timestep$$

$$netGainCumul = \sum_{t = BegMonth}^{d} \text{Heron Inflow Values}(\text{Low Net EOD Gain}, t) \times Timestep$$

$$seepageCumul = \sum_{t = BegMonth}^{d} \text{Seepage}(t) \times Timestep$$

ELSE (high flow conditions)

- Set the *inflowRatio* for the day index in the FOR loop equal to the low inflow ratio, and sum the cumulative inflow ratio, net gain and seepage from the beginning of the month to index day.

$$inflowRatio(d) = \text{Heron Inflow Values}(\text{High Ratio}, d)$$

$$inflowRatioNet = \sum_{t = BegMonth}^{d} \text{Heron Inflow Values}(\text{High Ratio}, t) \times Timestep$$

$$netGainCumul = \sum_{t = BegMonth}^{d} \text{Heron Inflow Values}(\text{High Net EOD Gain}, t) \times Timestep$$

$$seepageCumul = \sum_{t = BegMonth}^{d} \text{Seepage}(t) \times Timestep$$

END IF

Set the previous logic to the current logic to use for the next timestep

$$previousTimestepLogic = currentLogic$$

Calculate the monthly sum of each column in the Heron Inflow Values table series slot

For the final day of the month at a daily timestep, calculate the monthly sum of each column in the Heron Inflow Values table series slot. Flow values will be summed to the corresponding volumes. Set the sums at the final day of each month in the “Heron Inflow Totals” table series slot. For a monthly timestep, this slot will be empty; the monthly (end of timestep) values will be displayed in "Heron Inflow Values.”

Heron Inflow Totals

The “Heron Inflow Totals” slot holds the monthly sum of the“Heron Inflow Values” slot.

Set inflowRatioCumul and adjust netGainCumul if inflowRatioMonthly is negative

These values will be used to calculate the cumulative natural flow based on the current logic. *inflowRatioMonthly* is the cumulative ratio inflow for the current day (current controller timestep) passed to this function from the Heron Inflow method above.

IF $d = 1$ (first day of the month)

IF $inflowRatioMonthly \leq 0$

$$inflowRatioCumul = 0$$

$$netGainCumul = netGainCumul + inflowRatio \times Timestep$$

ELSE

$inflowRatioCumul = inflowRatio \times Timestep$

END IF

ELSE (after the beginning of the month)

IF $inflowRatioMonthly \leq 0$

$inflowRatioCumul = 0$

$netGainCumul = netGainCumul + inflowRatioNet$

ELSE

$prevInflowRatioCumul = \text{Heron Cumulative Inflow Values}(\text{Ratio Inflow}, d - 1)$

$inflowRatioCumul = prevInflowRatioCumul + inflowRatio \times Timestep$

END IF

END IF

IF $netGainCumul < 0$

$netGainCumul = 0$

END IF

Set the cumulative natural flow based on the current (controller timestep) logic

The current logic (*currentLogic*) is passed to this function from the Heron Inflow method above.

IF $currentLogic = NetGain$

$inflowWithinCumul = netGainCumul$

$naturalFlowCumul = inflowWithinCumul$

ELSE IF $currentLogic = Seepage$

$inflowWithinCumul = seepageCumul$

$naturalFlowCumul = inflowWithinCumul$

ELSE IF $currentLogic = Ratio$

$inflowWithinCumul = 0$

$naturalFlowCumul = inflowRatioCumul$

ELSE IF $currentLogic = SeepageRatio$

$inflowWithinCumul = seepageCumul$

$naturalFlowCumul = inflowRatioCumul + inflowWithinCumul$

ELSE IF $currentLogic = NetGainRatio$

$$inflowWithinCumul = netGainCumul$$

$$naturalFlowCumul = inflowRatioCumul + inflowWithinCumul$$

END IF

Set the values in the Heron Cumulative Inflow Values table series slot at timestep d

---

Note: *The "Heron Cumulative Inflow Values" displayed below are an adjustment when the "findHeronRGInflow function" is used. This results in the overriding of the values that have already been set at timestep d.*

Heron Cumulative Inflow Values

Column	Value
Ratio Inflow (Volume)	inflowRatioCumul
Net EOD Gain (Volume)	netGainCumul
Inflow Within (Volume)	inflowWithinCumul
Natural Flow (Volume)	naturalFlowCumul

Set the RioGrande.Slot Inflow at timestep d to difference between the natural flow at this timestep and the RioGrande.Inflow at this timestep

IF $d = 1$ (first day of the month)

$$\text{RioGrande.Slot Inflow}(d) = \frac{naturalFlowCumul}{Timestep} - \text{RioGrande.Inflow}(d)$$

ELSE (after the first day of the month)

$$prevNaturalFlowCumul = \text{Heron Cumulative Inflow Values}(\text{Natural Flow}, d - 1)$$

$$\text{RioGrande.Slot Inflow}(d) =$$

$$\frac{(naturalFlowCumul - prevNaturalFlowCumul)}{Timestep} - \text{RioGrande.Inflow}(d)$$

END IF

At this point, the function will repeat with the next index day *d* until it has reached the current day (current controller timestep).

Sediment Transport

The Sediment Transport category is actually displayed on the Methods tab, not the Accounting Methods tab; however, this category is only available if the Abiquiu, Cochiti or Jemez Gain Loss method has been selected for the Reservoir Account Gain Loss category. (See “Abiquiu, Cochiti and Jemez Gain Loss” on page 205.) The sedimentation methods in this category are only called in Accounting. They are not called during physical simulation.

The method in this category—Total Vol Sed (Post2000) calculates the storage volume lost to sedimentation. See "Total Vol Sed (Post2000)."

Total Vol Sed (Post2000)

This method calculates the sediment load in tons as a function of the reservoir inflow. It then converts this to a volume based on the sediment density.

Note: *This method is hard-coded to calculate daily sediment load based on units of cfs for the reservoir inflow and tons for the sediment load. The Exponent value in the Seasonal Inflow Coeffs slot must be set accordingly.*

This method is currently configured to function correctly for daily and monthly timesteps only. At a monthly timestep, sediment load is calculated by first finding daily sediment load using the average daily inflow for the month the multiplying the daily load by the number of days in the month. The results from a monthly timestep may not exactly match the aggregate of the daily results for the month if the daily inflows vary significantly within the month.

The details of the individual components are provided in "Slots specific to the Total Vol Sed (Post2000) method." The steps carried out by this method are detailed in “Steps carried out by the Nambe Falls Gain Loss method” on page 247.

Slots specific to the Total Vol Sed (Post2000) method

Selecting the Total Vol Sed (Post2000) method instantiates the following slots:

+ Bed Load Coeffs

Type: Table Slot
Units: Flow, None, None
Description: Slope and intercept values for calculating Bed Load at discrete inflow values

The first column contains reservoir inflow values. The second column is the slope and the thrid column is the intercept used when calculating bed load for a given inflow. The slope and intercept should be based on daily sediment load in tons for an inflow in cfs, regardless of the timestep length or the units set on any other slot or column.The method uses a table lookup to select the slope and intercept value (not interpolation). It uses the coefficient and exponent from the row corresponding to the largest flow value than or equal to the reservoir Inflow

I/O: Required Input
Supply Links: No

+ Est Sed Deposition

Type: Table Series Slot
Units: Length, Mass, Mass, None, Volume, Volume
Description: The final outputs from the sediment calculations
The final column, Accumulated Perm Sediment, is the primary output value of this method. The value is also shown on the series slot Accumulated Permanent Sediment, which is used later by accounting methods. Details of the values in this table are given in the method description below.
I/O: Output
Supply Links: No

+ Permanent Sediment Content

Type: Table Slot
Units: Volume
Description: The storage volume occupied by permanent sediment

I/O: Required Input
Supply Links: No

+ Accumulated Permanent Sediment

Type: Series Slot
Units: Volume
Description: The accumulated storage volume occupied by permanent sediment. Values in this series slot are the same as the values in the Est Sed Deposition slot, Accumulated Perm Sediment column.
If the initial timestep value on this slot is not known, but the value on the table series slot is known, the table series slot value is copied to the series slot at the initial timestep and set as an input.
I/O: Initial Input, remaining values output.
Supply Links: No

+ Seasonal Inflow Coeffs

Type: Table Slot
Units: None, Flow, None, None
Description: The coefficient and exponent to use in calculating sediment load for a given month and a given reservoir inflow

The first column contains the month, represented by an integer. At least one row is required for each month. The second column contains inflow values, which should be monotonically increasing for each month. The third column contains coefficient values for each month and specified flow. The fourth column contains exponent values for each month and specified flow. The coefficient and exponent should be based on daily sediment load in tons for an inflow in cfs, regardless of the timestep length or the units set on any other slot or column. The method uses a table lookup to select the coefficient and exponent value (not interpolation). It uses the coefficient and exponent from the row corresponding to the largest flow value for the month less than or equal to the reservoir Inflow.

I/O: Required Input
Supply Links: No

+ Sed Data

Type: Table Slot
Units: None, None, Length, None, None, Sed Density
Description: Parameter values used for converting sediment load in tons to an equivalent volume

I/O: Required Input
Supply Links: No

Steps carried out by the Total Vol Sed (Post2000) method

The Total Vol Sed (Post2000) methodcarries out the following steps:

1. Calculate the daily inflow volume in *cfs-day*. See “Get the Bed Load Coefficients” on page 280.
2. Get the Bed Load Coefficients. See “Get the Bed Load Coefficients” on page 280.
3. Calculate the Bed Load. See “Calculate Bed Load in tons” on page 280.
4. Get the Seasonal Inflow Coefficients. See “Get the Seasonal Inflow Coefficients” on page 280.
5. Calculate the total sediment. See “Calculate the total sediment” on page 280.
6. Calculate the elevation of the permanent sediment pool. See “Calculate the elevation of the permanent sediment pool” on page 280.
7. Calculate the sediment above the perm pool. See “Calculate the sediment above the perm pool” on page 281.
8. Calculate the accumulated permanent sediment. See “Calculate the accumulated permanent sediment” on page 281.
9. Set all final values in the EST SED Deposition table serie slot. See “Set all final values in the EST SED Deposition table series slot” on page 282.
10. Set the accumulated permanent sediment. See “Set the Accumulated Permanent Sediment” on page 283.

Calculate the daily inflow volume in cfs-day

$$dailyInflowVol[cfs - day] = \text{Inflow}[cfs] \times Timestep[days]$$

IF isMonthly

For a monthly timestep the timestep inflow volume must be divided by the number of days in the month to get the daily inflow volume.

$$dailyInflowVol[cfs - day] = \frac{dailyInflowVol[cfs - day]}{daysInMonth}$$

END IF

Get the Bed Load Coefficients

Get the Bed Load Coefficients (*bedLoadSlope* and *bedLoadIntcpt*) from the Bed Load Coeffs table slot, corresponding to reservoir Inflow. Use the row for the largest Inflow value less than or equal to the reservoir Inflow.

Calculate Bed Load in tons

$$bedLoad[tons] = bedLoadSlope \times dailyInflowVol[cfs - day] + bedLoadIntcpt$$

Get the Seasonal Inflow Coefficients

Get the Seasonal Inflow Coefficients (*sedLoadCoeff*, and *sedLoadExpon*). Use the row for the largest Inflow value less than or equal to the reservoir Inflow.

Calculate the total sediment

11. Calculate total sediment load in tons.

$$totalSedLoad[tons] = sedLoadCoeff \times dailyInflowVol[cfs - day]^{sedLoadExpon} + bedLoad[tons]$$

IF isMonthly

$$bedLoad = bedLoad \times daysInMonth$$

$$totalSedLoad = totalSedLoad \times daysInMonth$$

END IF

Calculate the elevation of the permanent sediment pool

12. Calculate the elevation of the permanent sediment pool

This calculation will be different for Cochiti than for Abiquiu and Jemez. For Cochiti the calculation depends on whether the Time Varying Elevation Volume method has been selected for the Sediment category. The previous Accumulated Perm Sediment will be added to the sediment content when calculating the *permPoolElev* for Cochiti unless the Time Varying Elevation Volume method is selected *and* the current timestep matches the date in a column header of the Elevation Volume Table Time Varying slot (*isElevVolModDate* = TRUE).

$$prevAccumPermSed = \text{Accumulated Permanent Sediment(t-1)}$$

IF *Reservoir* = Abiquiu OR *Reservoir* = Jemez

$$permPoolElev = TableInterpolation(\text{Elevation Volume Table, Permanent Sediment Content})$$

ELSE IF *Reservoir* = Cochiti

IF (Time Varying Elevation Volume is selected) AND (*isElevVolModDate* = TRUE)

$$permPoolElev = TableInterpolation(\text{Elevation Volume Table, Permanent Sediment Content})$$

ELSE

$$tempSedStorage = \text{Peramanent Sediment Content} + prevAccumPermSed$$

$$permPoolElev = TableInterpolation(\text{Elevation Volume Table, } prevAccumPermSed)$$

END IF

END IF

Calculate the sediment above the perm pool

Calculate the sediment above the perm pool (a percent) and the volume lost to sedimentation at this timestep.

IF $\text{Pool Elevation} < permPoolElev$

$$sedAbovePermPool = 0$$

ELSE

$$zeroStorElev = \text{Sed Data(Zero Storage Elev)}$$

$$percSedConst = \text{Sed Data(Percent Sed Const)}$$

$$percSedExpon = \text{Sed Data(Percent Sed Expon)}$$

$$sedAbovePermPool = \left(\frac{\text{Pool Elevation} - permPoolElev}{(permPoolElev - zeroStorElev) \times percSedConst}\right)^{percSedExpon}$$

END IF

$$volumeLost = \frac{\text{Sed Data(Trap Efficiency)} \times totalSedLoad \times (1 - 0.01 \times sedAbovePermPool)}{\text{Sed Data(Sed Density)}}$$

Calculate the accumulated permanent sediment

Calculate the accumulated permanent sediment as the previous accumulated sediment plus the volume lost to sedimentation at the current timestep, unless the Time Varying Elevation Volume method is selected and it is an ElevVol Mod Date.

IF (Time Varying Elevation Volume is selected) AND (*isElevVolModDate* = TRUE)

$$accumPermSed = 0$$

ELSE (the typical case)

$$accumPermSed = prevAccumPermSed + volumeLost$$

END IF

Set all final values in the EST SED Deposition table series slot

Set all final values in the Est Sed Deposition table series slot.

Est Sed Deposition

Column Header (with unit type	**Value**
Perm Pool Elev (Length)	permPoolElev
Bed Load (Mass)	bedLoad
Total Load (Mass)	totalSedLoad
% Sed abv Perm pool (None)	sedAbovePermPool
Total Pool Sediment (Volume)	volumeLost
Accumulated Perm Sediment (Volume)	accumPermSed

Set the Accumulated Permanent Sediment

Set the slot Accumulated Permanent Sediment = *accumPermSed*

Stream Gage Account Slot Inflow

The stream gage has three Rio Grande specific methods. See

- "Reconcile Rio Grande."
- "All San Juan Chama."
- "All Rio Grande."

Reconcile Rio Grande

This method computes the total accounting Slot Inflow to reconcile the physical and accounting flows based on the water type of the accounts. Accounts with water type “SanJuan” have their Slot Inflows set to 0.0cfs. The remaining total reconciliation water is then proportioned to the Slot Inflow on accounts with the RioGrande water type.

The logic is as follows:

```
Get a list of all the Inflow slots on accounts with water type RioGrande. and register each as a dependency

RioGrandeInflowSUM = Sum of RioGrande Inflows

Get a list of all the Slot Inflow slots on accounts with water type RioGrande.

Get a list of all the Slot Inflow slots on accounts with water type SanJuan.

Get a list of all the Outflow slots on accounts with water type SanJuan and register each as a dependency

SanJuanOutflowSUM = Sum of SanJuan Outflows

Get the StreamGage.Gage Outflow
```

$$\text{RioGrandeSlotInflowTOTAL} = \text{StreamGage.GageOutflow} - \text{SanJuanOutflowSUM} - \text{RioGrandeInflowSUM}$$

```
If the RioGrandeInflowSUM is zero, then divide any Slot Inflows equally to all Rio Grande accounts.

Otherwise, for each account with water type RioGrande:
```

$$\text{account.Slot Inflow} = \frac{\text{account.Inflow}}{\text{RioGrandeInflowSUM}} \times \text{RioGrandeSlotInflowTOTAL}$$

```
Set Slot Inflows on accounts with water type SanJuan to 0.0.
```

All San Juan Chama

All RioGrande account Slot Inflows are set to 0.0. The Gage Outflow is then divided evenly by the number of SanJuan accounts and set on each SanJuan Slot Inflow.

All Rio Grande

All SanJuan account Slot Inflows are set to 0.0. The RioGrande Slot Inflow is divided evenly by the number of RioGrande accounts and set on each accounts Slot Inflow slot.