



**CADSWES**

University of Colorado

Center for Advanced Decision Support for Water and Environmental Systems

# Water Accounting Modeling

---

RiverWare User Group Meeting  
February 6 - 7, 2007

# Presentation Outline

---

## ➤ Water Accounting Capabilities

- Water Accounting Motivation and Definitions
- Accounts
- Supplies
- Solution Algorithm
- User Defined Accounting Methods
- Exchanges
- Demonstration as we go....

## ➤ Demonstration of Water Accounts Summary utility

## ➤ Accounting Class Outline

## ➤ Prioritized Water Rights Modeling and Allocation

# Motivation

---

- Basins are operated for many purposes
- In many basins, it is necessary to track not only volume of water but the ownership and type
- Operating decisions are dependent on a user's available water, legal restrictions, physical constraints, and exchange mechanism
- Priority water rights allocation

- 
- A framework is necessary to model water type and ownership

**Solution:**

# Water Accounting

# Requirements for Water Accounting

---

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

# How is water accounting modeled 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.

# “Physical” vs. “Paper” water in RiverWare

---

## ➤ Physical (wet) Water:

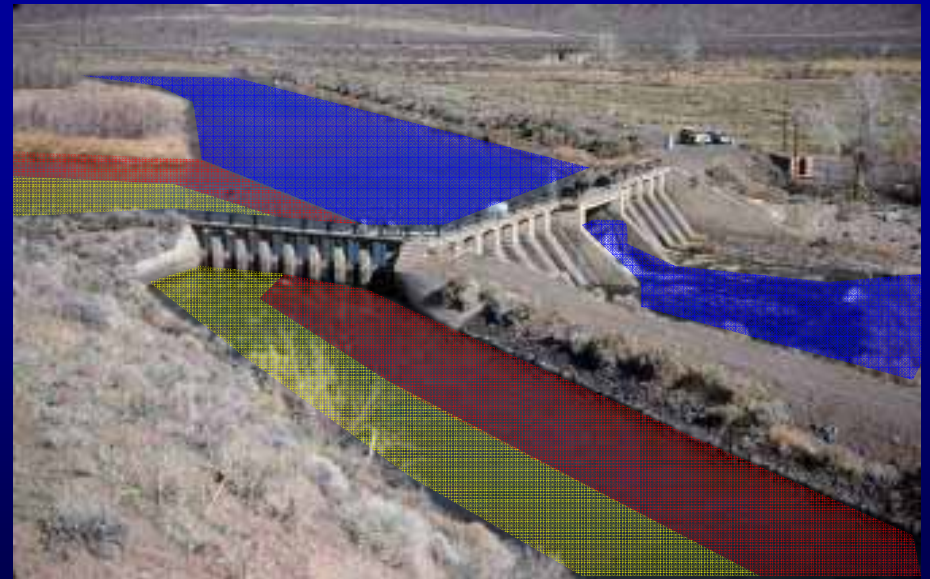
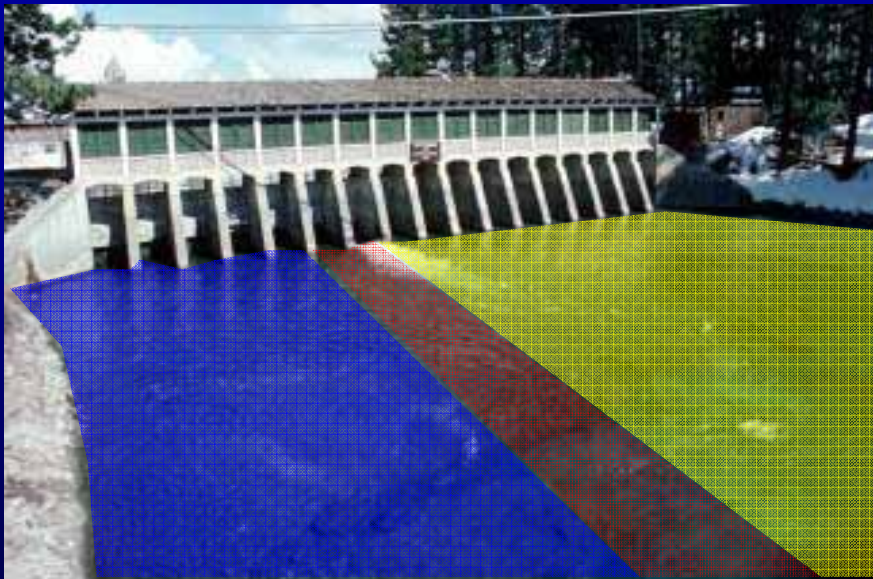
- Total volume/flow of water in a river basin object. For example, storage in a reservoir represents the volume of physical water in the reservoir



# “Physical” vs. “Paper” water in RiverWare

---

- Paper (colored) Water - type and ownership:
  - Volume/flow of water classified by type or ownership. For example, a certain agency owns 500AF of physical water in the reservoir.





# Types of Accounts

---

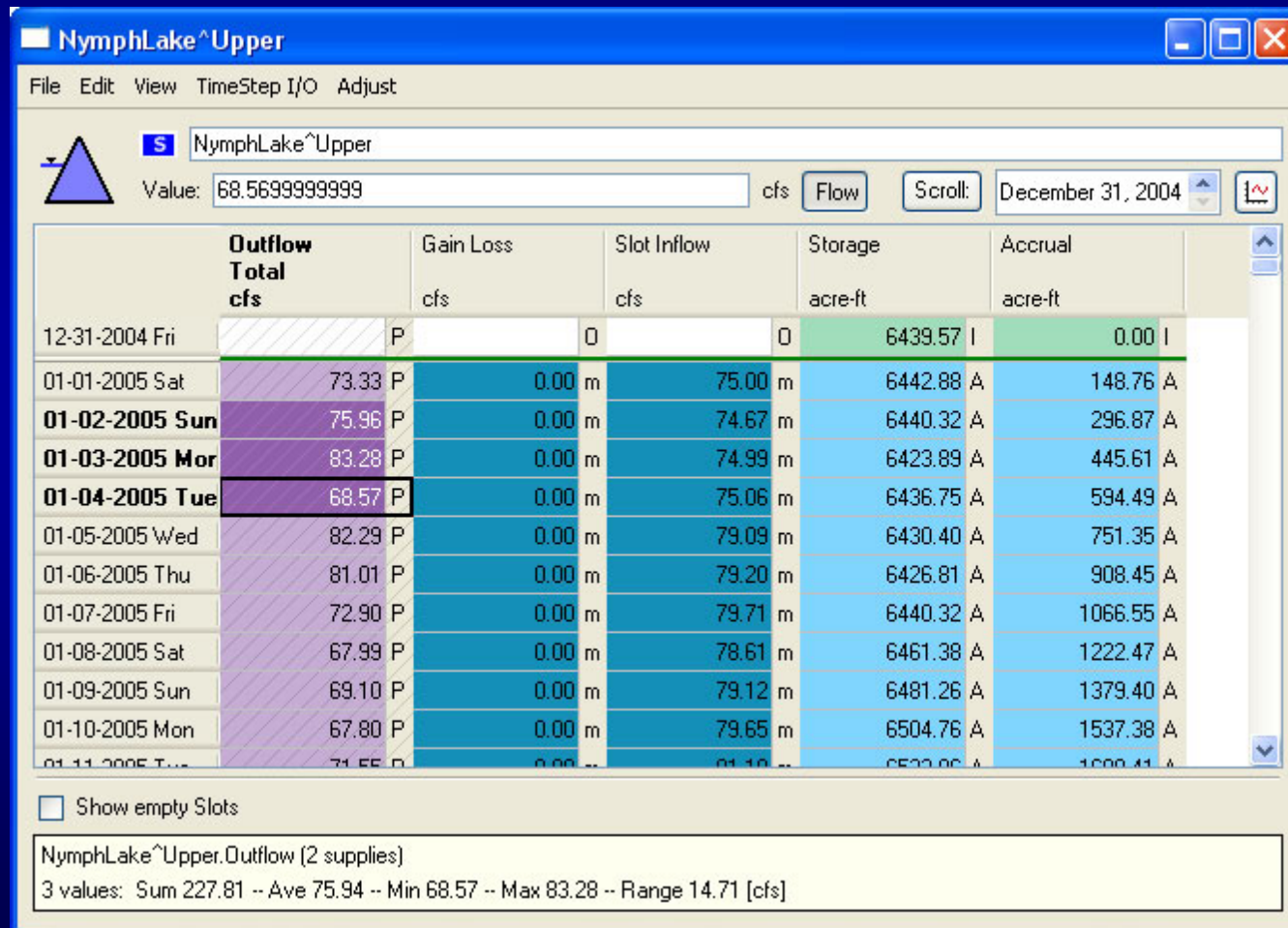
## ➤ Legal Accounts:

- Storage Account - on Reservoirs
  - Storage, gain/loss, accrual, carry-over, inflow and outflow, transfers
- Diversion Account - on Water Users, Aggregate Diversions
  - diversion, consumption, and return flow
- Instream Flow – Track total flow into all accounts

## ➤ Non-Legal Accounts

- Pass-through Accounts - Reaches, Confluences, Gages, Reservoirs, etc...
  - keep track of water moving between legal accounts

# Storage Account



# Storage Account – Mass Balance

---

- $$\text{Storage}(t) = \text{Storage}(t-1) + \text{GainLoss}(t) + \text{TimestepLength} * (\text{Inflow}(t) - \text{Outflow}(t) + \text{SlotInflow}(t) - \text{Diversion}(t) - \text{ReturnFlow}(t) + \text{TransfersIn}(t) - \text{TransfersOut}(t))$$
- $$\text{Accrual}(t) = \text{Accrual}(t-1) + \text{TimestepLength} * (\text{Inflow}(t) + \text{SlotInflow}(t))$$
  - Required Knowns: Storage(t-1), Accrual(t-1), Outflow(t), Slot Inflow(t), Gain Loss(t)
  - Note: if it is the begin accrual date, use BeginYearAllocation(t) in place of Storage(t-1) and add in Carryover(t)

# Diversion Account

RedRiverUsers^RedRiverUsers

File Edit View TimeStep I/O Adjust

Value: 5 cfs Flow Scroll: cember 31, 2004

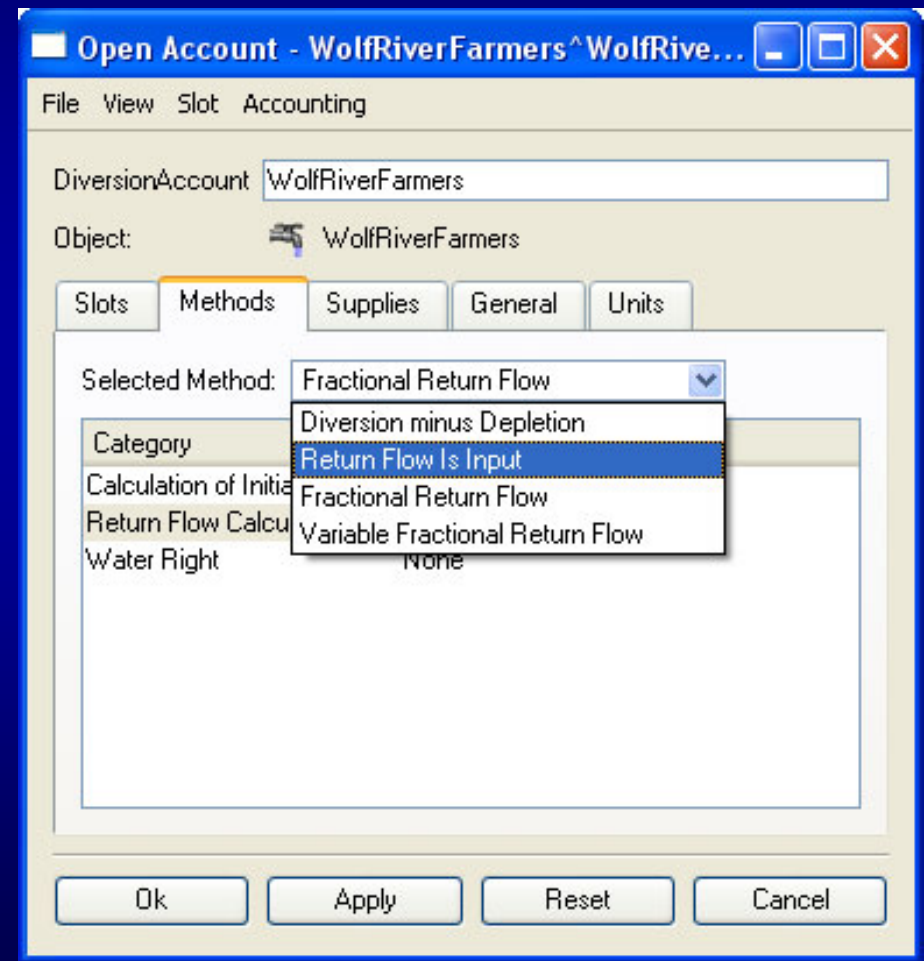
	Diversion Total cfs	Depletion cfs	returnFlow cfs
02-11-2005 Fri	5.00	3.75 A	1.25
02-12-2005 Sat	5.00	3.75 A	1.25
<b>02-13-2005 Sun</b>	5.00	3.75 A	1.25
<b>02-14-2005 Mon</b>	5.00	3.75 A	1.25
<b>02-15-2005 Tue</b>	5.00	3.75 A	1.25
<b>02-16-2005 Wed</b>	5.00	3.75 A	1.25
02-17-2005 Thu	5.00	3.75 A	1.25
02-18-2005 Fri	5.00	3.75 A	1.25

Show empty Slots

RedRiverUsers^RedRiverUsers.Diversion (RedRiverDiversion)  
4 values: Sum 20.00 -- Ave 5.00 -- Min 5.00 -- Max 5.00 -- Range 0.00 [cfs]

# Diversion Account – Mass Balance

- Depends on selected return flow method
- Depends on knowns and unknowns



# Diversion Account – Mass Balance

---

## ➤ Diversion minus Depletion

- $\text{ReturnFlow} = \text{Diversion} - \text{Depletion}$

OR

- $\text{Depletion} = \text{Diversion} - \text{ReturnFlow}$

- Also:  $\text{Accrual} = \text{Accrual}(t-1) + (\text{Depletion})(\text{TimestepLength})$

## ➤ Return Flow is Input

- $\text{Depletion} = \text{Diversion} - \text{ReturnFlow}$

- $\text{Accrual} = \text{Accrual}(t-1) + (\text{Depletion})(\text{TimestepLength})$

# Diversion Account – Mass Balance

---

## ➤ Fractional Return Flow

- $\text{ReturnFlow} = (\text{Diversion})(\text{FractionOfDiversion})$
- $\text{Depletion} = \text{Diversion} - \text{ReturnFlow}$
- $\text{Accrual} = \text{Accrual}(t-1) + (\text{Depletion})(\text{TimestepLength})$

## ➤ Variable Fractional Return Flow

- $\text{ReturnFlow} = (\text{Diversion})(\text{VariableFractionOfDiversion})$
- $\text{Depletion} = \text{Diversion} - \text{ReturnFlow}$
- $\text{Accrual} = \text{Accrual}(t-1) + (\text{Depletion})(\text{TimestepLength})$

# Instream Flow Account

ControlPoint0^InstreamFlowAccount0

File Edit View TimeStep I/O Adjust

ControlPoint0^InstreamFlowAccount0

Value: 26 cfs Flow Scroll: December 31, 2004

	Inflow Total cfs	Outflow cfs	Accrual m3	Maximum Accrual m3	Flow cfs	
12-31-2004 Fri		0	0	0	0	P
<b>01-01-2005 Sat</b>	1.00 I	1.00 m	0	0	7.00 A	
01-02-2005 Sun	1.00 I	1.00 m	0	0	7.00 A	
01-03-2005 Mon	1.00 I	1.00 m	0	0	8.00 A	
01-04-2005 Tue	1.00 I	1.00 m	0	0	7.00 A	
01-05-2005 Wed	1.00 I	1.00 m	0	0	7.00 A	

Show empty Slots

ControlPoint0^InstreamFlowAccount0.Flow  
1 value: 7.00 [cfs]



# Instream Account – Mass Balance

---

- $\text{Outflow} = \text{Inflow}$
- $\text{Accrual} = \text{Accrual}(t-1) + (\text{Inflow})(\text{TimestepLength})$
- $\text{Flow} = \Sigma (\text{All account inflows into object})$

# Passthrough Account

LeakyReach^Fish

File Edit View TimeStep I/O Adjust

Value:  Flow Scroll: December 31, 2004

	Inflow Total cfs	Outflow cfs	Gain Loss cfs	Slot Inflow cfs
<b>12-31-2004 Fri</b>	0	0	0	0
01-01-2005 Sat	100.00	76.00 A	-24.00 m	0.00 m
01-02-2005 Sun	100.00	76.00 A	-24.00 m	0.00 m
01-03-2005 Mon	80.00	56.00 A	-24.00 m	0.00 m

Show empty Slots

4 Slots [@ 24:00 December 31, 2004]  
0 values:

# Passthrough Account – Mass Balance

---

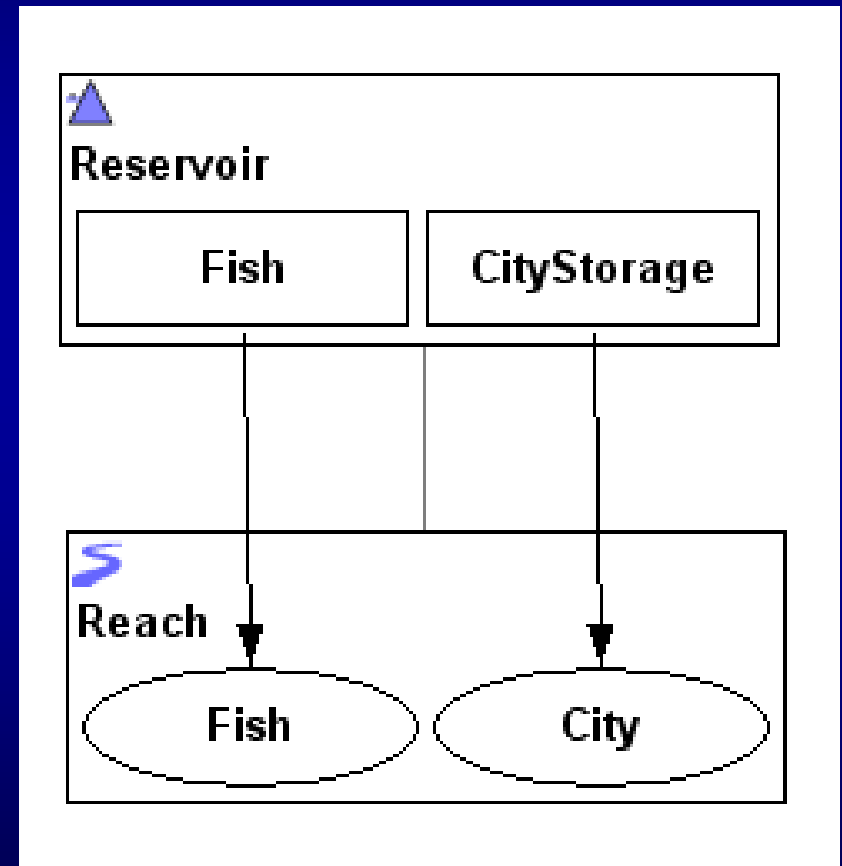
## ➤ General equation:

- $\text{Outflow} = \text{Inflow} + \text{Slot Inflow} - \text{Gain Loss} - \text{Diversion} + \text{Return Flow} + \text{Transfers In} - \text{Transfers Out}$
- Required knowns: Either Inflow, Gain Loss, or Slot Inflow must be known for the account to solve. If none are known, the account will not solve. If one or two of these slots are known, the other(s) default to zero

## ➤ On reservoirs, temporary storage can be allowed when operating constraints preclude releasing all inflows

# Linking Accounts

- Accounting network is formed by links between account called “supplies”
- **Supply definition:**  
A supply is used to link two accounts. Specifically, a supply to an account means that paper water is moving into that account from another (often upstream) account.



# Optional Properties of Accounts and Supplies

---

## ➤ Account

- Water Type – For example, trans-basin water or local water
- Water Owner – For example, Contractor A, Contractor B, and City

## ➤ Supply

- Type – Inflow/Outflow, Diversion/Return Flow, Transfer
- Destination – For example, Diversion A, Red Reach
- Release Type – For example, Diversion, Fish

# Accounting Solution

---

## ➤ “Spreadsheet” type solution

- Account solves when it has the required knowns
- Account solves whenever it gets a new value in any of the mass balance components (not controlled by dispatcher during a run)
- Account slots and supplies are set by user input or rules

# Accounting Solution

---

## ➤ Mass balance solutions:

- Storage accounts solve only for storage balance (user specifies storage account outflows)
- Passthrough accounts solve only downstream for outflow
- Diversions accounts solve for depletion and/or return flow
- Instream Flow accounts solve for
  - Flow (Total accounting flow into object)
  - Outflow

# Object Level Accounting Methods

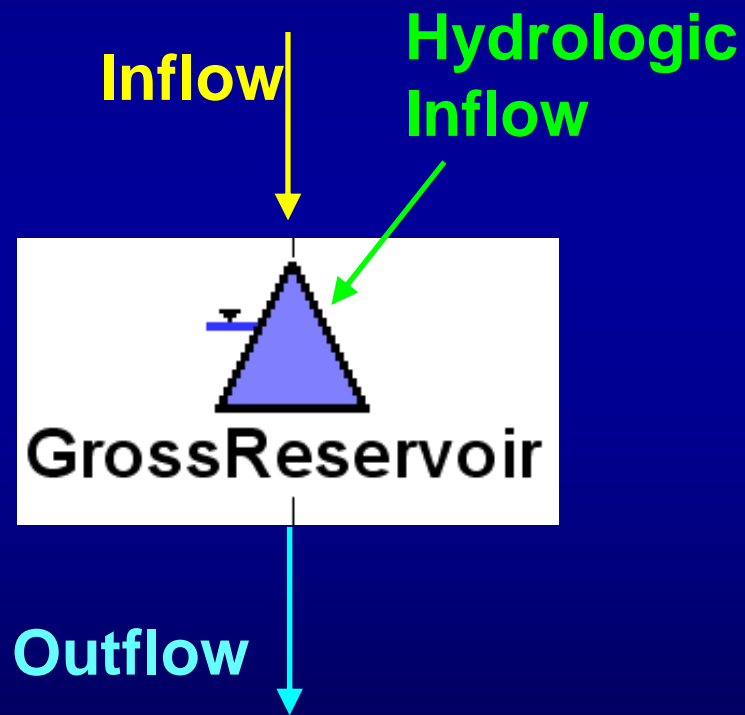
---

- Purpose: Distribute physical water on simulation objects to the accounts
- Methods are on the object and apply to all accounts
- Categories
  - Gain Loss: allocate physical gains and losses
  - Slot Inflow: allocate local inflows
  - Reconciliation: have accounting system match physical
- Two ways to specify the methods
  - Compiled: Simple methods or basin specific
  - User Defined: Written in the RiverWare Policy Language (RPL)

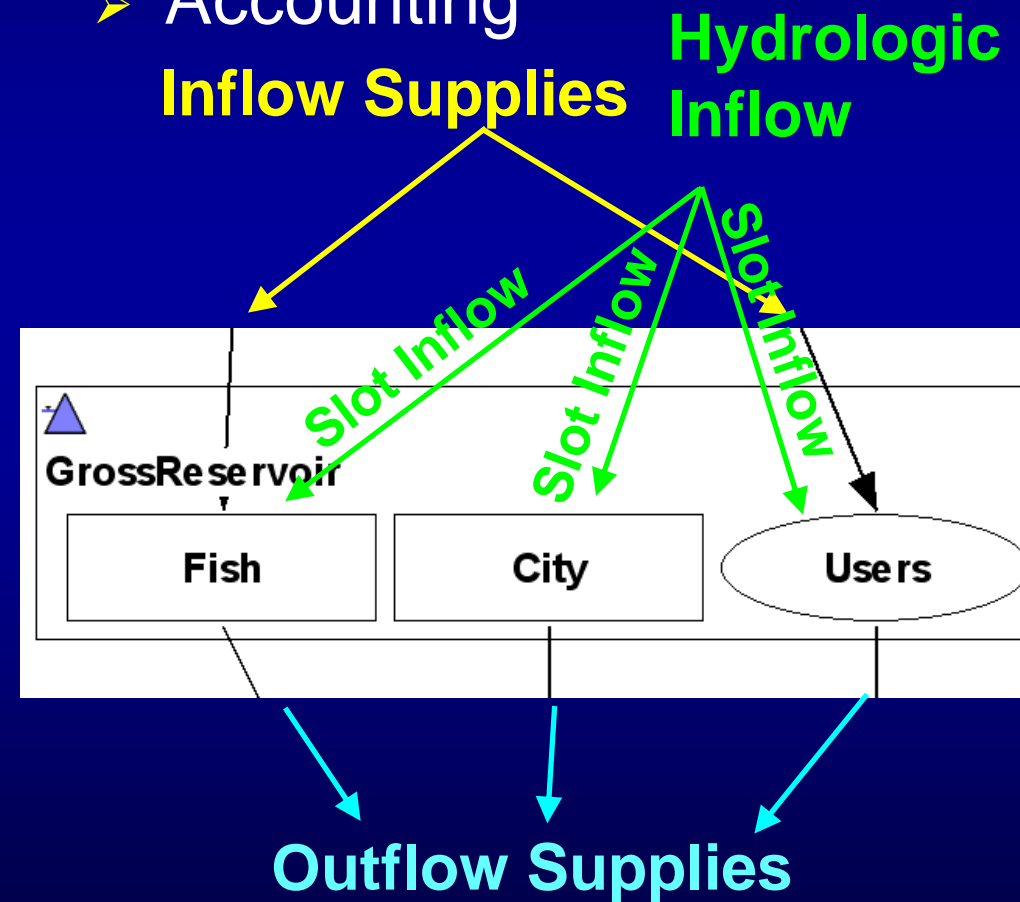


# Slot Inflow Example

## ➤ Physical



## ➤ Accounting



# Slot Inflow Example (cont.)

The image shows two overlapping windows from the RiverWare software. The left window is titled "Method Set Editor - 'Object-level Accounting Methods'". It contains a table of methods with columns for Name, Priority, On, and Type. The right window is titled "Open Object - GrossReservoir" and shows a dropdown menu for "Selected Method:" with "GrossReservoir Slot Inflow" selected. An arrow points from the "StorageReservoir" method in the left window to the selected method in the right window.

**Method Set Editor - "Object-level Accounting Methods"**

Name	Priority	On	Type
Conin PassThrough Slot Inflow		✓	Category Group
Pipe Junction Pass Through Slot Inflow		✓	Category Group
Pipe PassThrough Slot Inflow		✓	Category Group
Inline Pump Pass Through Slot Inflow		✓	Category Group
DistributionCanal PassThrough Slot Inflow		✓	Category Group
Storage Account Slot Inflow		✓	Category Group
StorageReservoir	1	✓	Method
Storage Account Gain Loss		✓	Category Group
Reservoir Reconciliation		✓	Category Group
Control Point Pass Through Slot Inflow		✓	Category Group
Water User Reconciliation		✓	Category Group
Gage PassThrough Slot Inflow		✓	Category Group
Reach PassThrough Slot Inflow		✓	Category Group
Pass Through Acct Gain Loss		✓	Category Group

**Open Object - GrossReservoir**

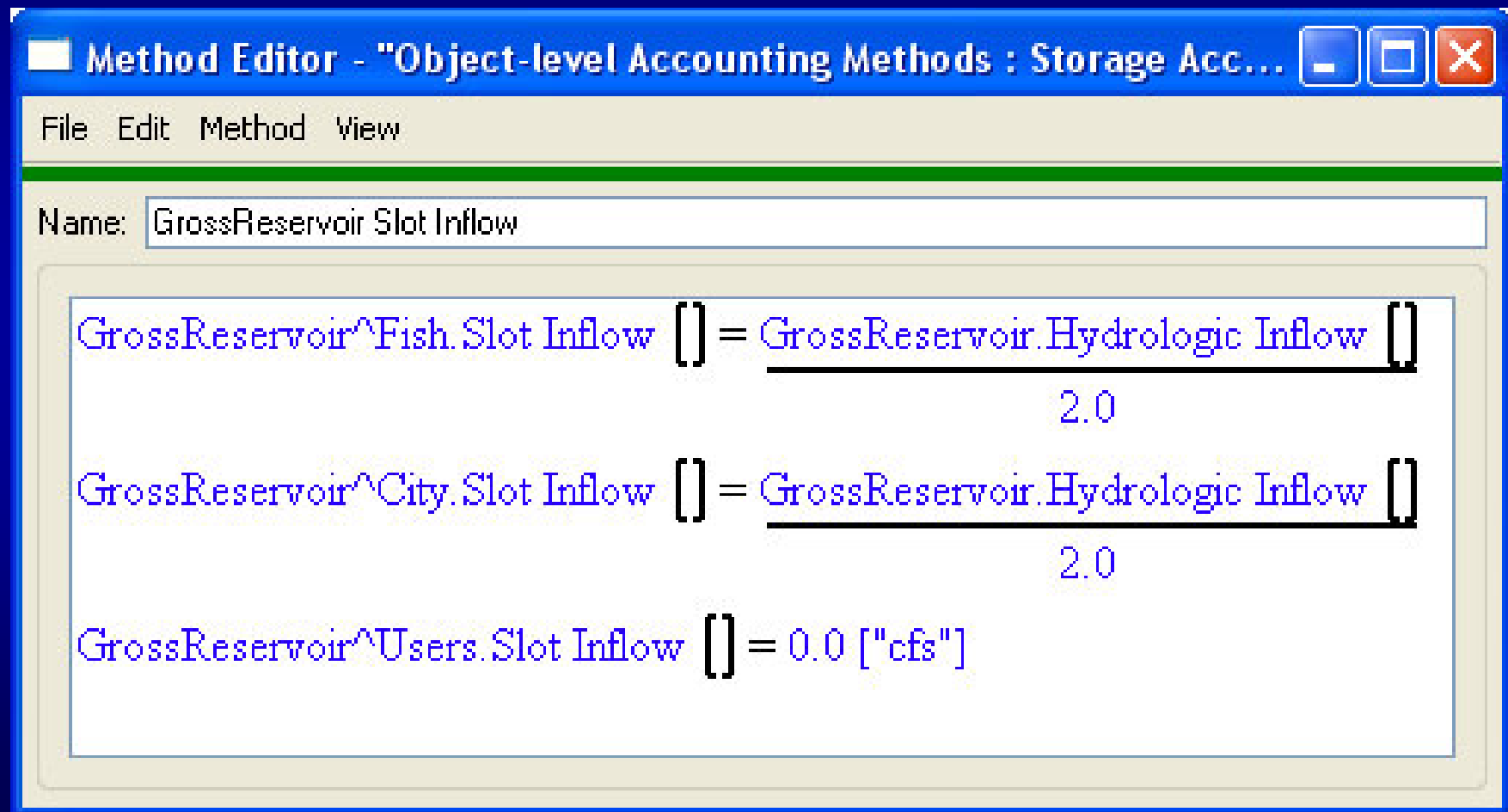
Object Name: GrossReservoir

Selected Method: GrossReservoir Slot Inflow

Category

- No Accounting Inflow Calculation
- Zero Slot Inflows
- Heron Inflow Calculation
- Pooled Account Slot Inflow
- Donner Inflow
- Prosser Uncomm
- GrossReservoir Slot Inflow**
- No Uncertainty
- No Loss Calc
- GrossReservoir Slot Inflow

# Slot Inflow Example (cont.)



The screenshot shows a window titled "Method Editor - 'Object-level Accounting Methods : Storage Acc...". The window has a menu bar with "File", "Edit", "Method", and "View". Below the menu bar is a "Name:" field containing "GrossReservoir Slot Inflow". The main area contains three equations:

```
GrossReservoir^Fish.Slot Inflow [] = GrossReservoir.Hydrologic Inflow [] / 2.0
GrossReservoir^City.Slot Inflow [] = GrossReservoir.Hydrologic Inflow [] / 2.0
GrossReservoir^Users.Slot Inflow [] = 0.0 ["cfs"]
```

# Use Rules to Control Accounting

---

Rules can be used to set supplies (links between accounts)

For example:

```
FishSupplyName.Supply [ ] =  
    FlowTarget() -  
    SumOtherReservoirReleasesOfType("Fish")
```

If account Slot Inflow and Gain Loss are known, this causes the account to solve for Storage

# Predefined functions for Accounting

---

- Many predefined functions specific to Accounting, for example:
  - AccountNamesByWaterType
  - DestinationsFromObjectReleaseType
  - ObjectsFromAccountName
  - SumAccountSlotsByWaterType
  - WaterOwners
  - ....
- Useful to generalize rule writing

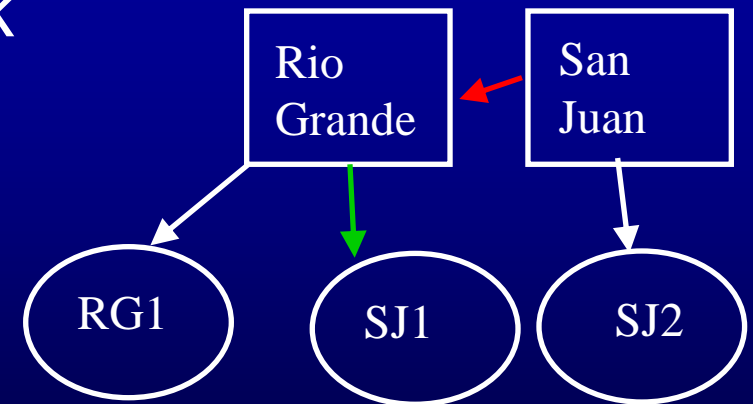
# Reconciliation

---

- Total physical releases/storage does not have to equal accounting releases/storage
- Matching the accounting system to the physical system is up to the user using:
  - Object Level Accounting Methods
  - Rules

# Water Exchanges

- Track water exchanges and transfers
- **Borrow** - a supply to an account or a user input
- **Source Payback** - an outflow (demand) from an account
- Destination Payback (optional) - a supply to an account
- In the exchange utility, the debt is updated whenever a new value is set in the borrow or payback
- Rules can access the debt using predefined functions



# Exchange Manager Utility

The screenshot displays two overlapping windows from the Exchange Manager Utility. The 'Exchange Configuration' window on the left shows settings for an exchange named 'Fish To Farmers'. It includes tabs for 'Main', 'Paybacks', and 'Units'. Under 'Borrow Supply', the 'Select Supply' radio button is chosen, and 'FishtoFarmerExchange' is selected in the dropdown. Under 'Payback Destination Supply', the 'No Destination' radio button is chosen. The 'Exchange Balance -- Fish To Farmers' window on the right shows a table of daily balances from February 18, 2005, to March 6, 2005. The table has three columns: 'Borrow (FishtoFarr acre-ft)', 'Source Balance acre-ft', and 'Debt(Fish to Farmer acre-ft)'. The 'Borrow' column shows a constant value of 9.92 (marked with 'R') from Feb 18 to Feb 28. The 'Source Balance' and 'Debt' columns show values marked with 'M' that increase over time.

**Exchange Configuration**

Main | Paybacks | Units

Exchange Name: Fish To Farmers

Borrow Supply:  Select Supply  Input Borrowed Amounts  
FishtoFarmerExchange

Payback Destination Supply:  Select Destination  No Destination

OK Apply Reset Cancel

**Exchange Balance -- Fish To Farmers**

File Edit System Config

**Fish To Farmers**  
**FishtoFarmerExchange**

	Borrow (FishtoFarr acre-ft)	Source Balance acre-ft	Debt(Fish to Farmer acre-ft)
February 18, 2005	R 9.92	M 178.51	M 198.35
February 19, 2005	R 9.92	M 188.43	M 209.37
February 20, 2005	R 9.92	M 198.35	M 220.39
February 21, 2005	R 9.92	M 208.26	M 231.40
February 22, 2005	R 9.92	M 218.18	M 242.42
February 23, 2005	R 9.92	M 228.10	M 253.44
February 24, 2005	R 9.92	M 238.02	M 264.46
February 25, 2005	R 9.92	M 247.93	M 275.48
February 26, 2005	R 9.92	M 257.85	M 286.50
February 27, 2005	R 9.92	M 267.77	M 297.52
February 28, 2005	R 9.92	M 277.69	M 308.54
March 1, 2005		M 259.83	M 288.71
March 2, 2005		M 241.98	M 268.87
March 3, 2005		M 224.13	M 249.04
March 4, 2005		M 206.28	M 229.20
March 5, 2005		M 188.43	M 209.37
March 6, 2005		M 170.58	M 189.53

Source Balance



# Presentation Outline

---

- Water Accounting Capabilities
  - Water Accounting Motivation and Definitions
  - Accounts
  - Supplies
  - Solution Algorithm
  - User Defined Accounting Methods
  - Exchanges
  - Demonstration as we go....
- **Demonstration of Object Accounts Summary utility**
- Accounting Class Outline
- Prioritized Water Rights Modeling and Allocation

# Object Account Summary

- Ported to Qt
- Demo!

Object Account Summary - Sample Reservoir

File Edit View TimeStep I/O Config Adjust

Single Object: Sample Reservoir Select Object ...

Columns:
 

- Acct Slots
- TimeSteps

Account Types:
 

- Storage
- InstreamFlow
- Diversion
- PassThrough

Account Slot Entity: Storage

Accounts (3 of 7)  List All Accounts

Sum	Ord	Object	Account	Type	Water Type	Water Owner
<input checked="" type="checkbox"/>	1	Sample Reservoir	S Contractor1	Stor	SanJuan	Contractor1
<input checked="" type="checkbox"/>	2	Sample Reservoir	S Contractor2	Stor	SanJuan	Contractor2
<input checked="" type="checkbox"/>	3	Sample Reservoir	S Contractor3	Stor	SanJuan	Contractor3
<input type="checkbox"/>	4	Sample Reservoir	S FederalSanJuan	Stor	SanJuan	NONE
<input type="checkbox"/>	5	Sample Reservoir	S RioGrande	Stor	RioGrande	NONE
<input type="checkbox"/>	6	Sample Reservoir	S StorageAccount0	Stor	NONE	NONE
<input type="checkbox"/>	7	Sample Reservoir	S StorageAccount1	Stor	NONE	NONE

Select All Accounts Set Selection ↑ ↓ Set Order

Value: 2485.00495868 acre-feet Scroll: December 31, 1995

	Sample Reservoir -- SUM Storage	Sample Reservoir ^Contractor1 Storage	Sample Reservoir ^Contractor2 Storage	Sample Reservoir ^Contractor3 Storage
	acre-feet	acre-feet	acre-feet	acre-feet
12-31-1995 Sun	6500.00	2000.00 I	2000.00 I	2500.00 I
01-01-1996 Mon	6480.17	1996.03 A	1992.07 A	2492.07 A
<b>01-02-1996 Tue</b>	6464.30	1990.08 A	1988.10 A	2486.12 A
<b>01-03-1996 Wed</b>	6449.30	1982.15 A	1982.15 A	2485.00 A
01-04-1996 Thu	6433.43	1976.20 A	1980.17 A	2477.07 A

Show empty Slots

3 Slots  
6 values: Sum 12913.60 -- Ave 2152.27 -- Min 1982.15 -- Max 2486.12 -- Range 503.97 [acre-feet]

# Presentation Outline

---

- Water Accounting Capabilities
  - Water Accounting Motivation and Definitions
  - Accounts
  - Supplies
  - Solution Algorithm
  - User Defined Accounting Methods
  - Exchanges
  - Demonstration as we go...
- Demonstration of Water Accounts Summary utility
- Accounting Class Outline
- Prioritized Water Rights Modeling and Allocation

# Accounting Class Outline - Day 1

---

- Overview of accounting system and solution
- How accounting works
- Run and view an accounting model
  - Navigate accounting system using new accounting workspace (also through simulation workspace)
  - Become familiar with different elements of accounting system
  - Run, observe and analyze model output

# Accounting Class Outline - Day 2

---

- Rules and Accounting – Strategies and features
- “After-the-fact” Accounting model
  - Use rules to specify accounting releases and diversions from simulated flows
- Advanced RBS: Accounting Driven Simulation
  - Rulebased Simulation dependence on Accounting System
- Water Rights Allocation
  - Allocation based on account water right priority date

# Accounting Class Outline - Day 3

---

- **Group/Interactive Exercise: Building an Accounting Model**
  - Start with existing simulation model
  - Create accounting system according to specifications
  - Mix of white board session and work on the computer
  - Write rules to implement accounting policy

# Presentation Outline

---

- Water Accounting Capabilities
  - Water Accounting Motivation and Definitions
  - Accounts
  - Supplies
  - Solution Algorithm
  - User Defined Accounting Methods
  - Exchanges
  - Demonstration as we go...
- Demonstration of Water Accounts Summary utility
- Accounting Class Outline
- **Prioritized Water Rights Modeling and Allocation**

# Prioritized Water Rights Modeling and Allocation Outline

---

- Requirements
- Modeling Requests for Water
- Types of Water Rights Modeled
- General Description of Model
- Predefined Rule Function
- Solution Algorithm
- Status of Work
- Summary



# Requirements

---

- Each right has a priority date
- No two accounts may have the same priority date
- A right's allowed quantity of water may be described in various ways on legal documents; the system must be flexible enough to model these ways of defining rights
- First in time, first in right
  - A junior right may not "short" a senior right

# Requirements (cont'd)

---

- Three distinct types of rights must be accommodated:
  - The right to divert from a stream
  - The right to store water
  - The right to ensure minimum in-stream flow
- Some rights are subject to physical constraints that are defined in terms of the state of the network **after senior rights have been satisfied**
  - Physical capacity of diversion structure
  - Size of reservoir

# Requirements (cont'd)

---

- Some rights are subject to legal constraints that are defined in terms of the state of the network **after senior rights have been satisfied**
  - Minimum bypass requirements, a form of subordination to other rights
  - Accrual-based maxima on the rights
- The effects of an upstream allocation on a downstream right must take into account lags and losses in the stream

# Requirements (cont'd)

---

- Must improve performance over RPL-based implementations
- RiverWare must be able to replicate results of WAM (Texas WRAP model)

# Modeling Requests for Water

---

- Each right makes an *Initial Request* based on the legal description of the right
- The state of the system at the time of allocation might restrict the initial request by
  - Physical constraints
  - Legal constraintsto produce a net *Allocation Request*
- The difference between the *Initial Request* and the amount allocated is the *Shortage*

# Types of Rights Modeled

---

## ➤ Diversion Rights

- Right to divert from a stream (via a supply of type “diversion/return flow”)
- Modeled with a diversion account
- Initial allocation request may be user-input or disaggregated from an annual request
- Subject to physical capacity of diversion structure
- May be subject to other legal constraints

# Types of Rights Modeled (cont'd)

---

## ➤ Storage Rights

- Right to store water from a stream
- Modeled with a storage account
  - On-stream: supplied by a passthrough account on a reservoir (via a supply of type “transfer”)
  - Off-stream: supplied by a passthrough account on a diversion object (via a supply of type In/Out), which is in turn supplied by a passthrough account on a reach (supply type Div/Ret)
- Subject to physical capacity of reservoir; off-stream rights also subject to capacity of diversion structure
- Initial allocation request may be user-input or “fill conservation pool of reservoir”
- May be subject to other legal constraints

# Types of Rights Modeled (cont'd)

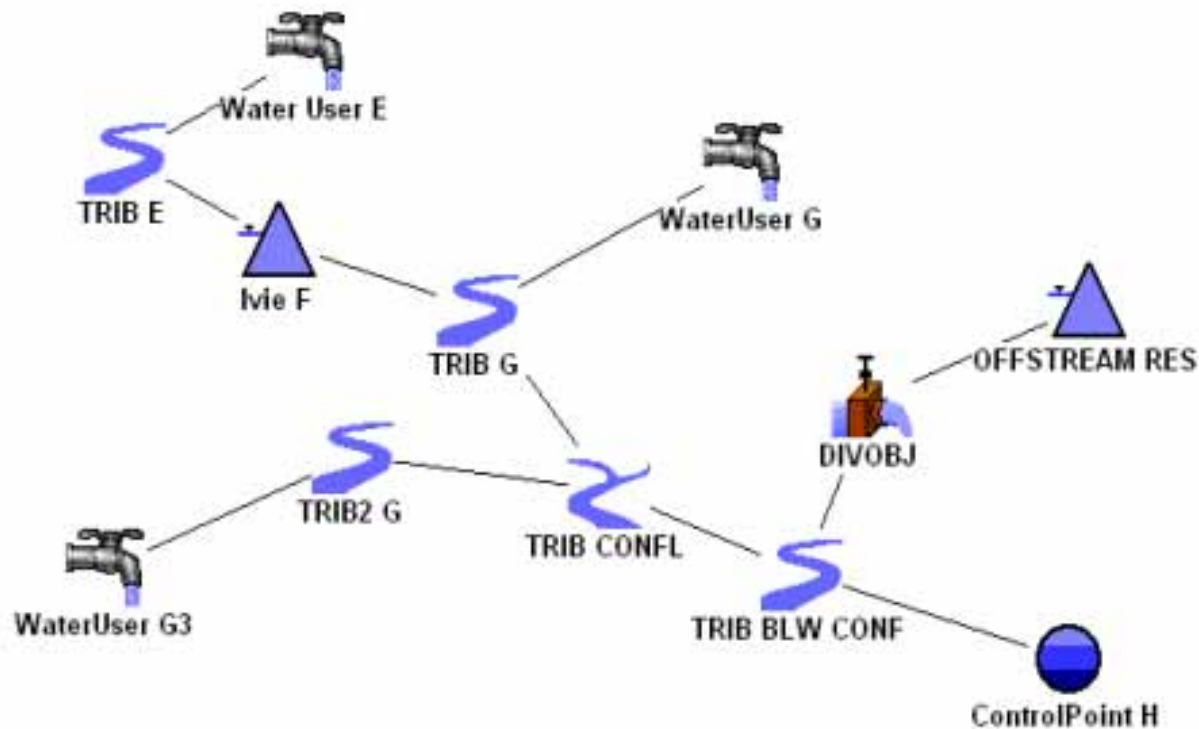
---

## ➤ Instream Flow Rights

- Right to keep water in a stream
  - Irrespective of the “color” of that water (natural flow or project water)
  - Nevertheless, limited to the amount of natural flow that is in the stream after senior rights are satisfied
- Modeled with an instream flow account on a control point object
- No supply required; the supply is implicit
- *Flow* slot shows total flow at control point
- Initial allocation request may be user-input or based on a reference level and seasonal mapping of reference level to flow
  - Reference level is computed by the control point, based on sum of reservoir storages at end of a reference year



# General Description of Model



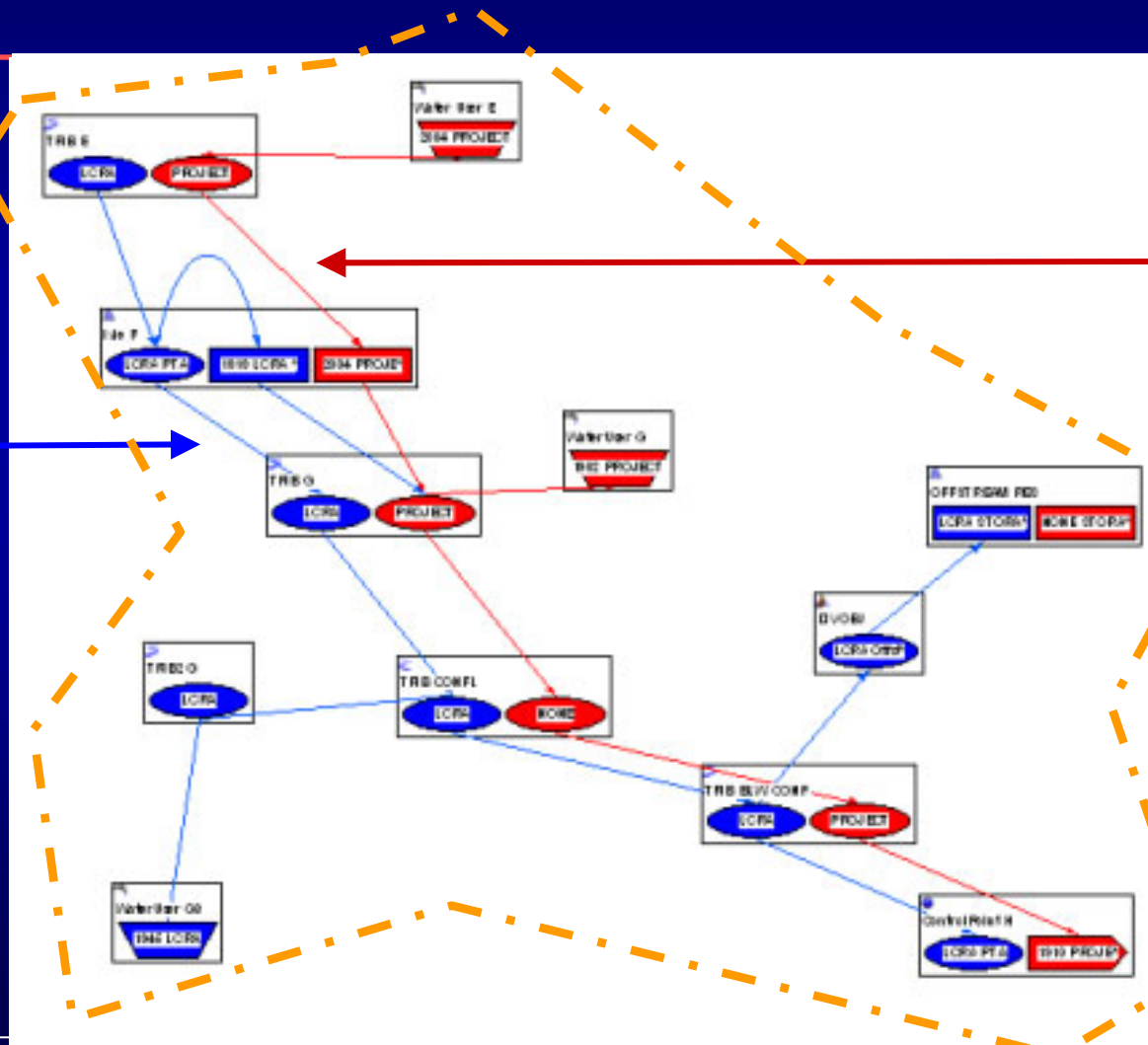
# General Description of Model (cont'd)

---

- Natural-flow supply chain is identified by water type of the supplying accounts
- Objects containing the water-rights accounts are members of a **Computational Subbasin** for the purpose of computing appropriations
- Computational Subbasin clones the accounts; all computations for the solution are performed in clone-world

# General Description of Model (cont'd)

Natural-Flow supply chain



Project-water supply chain

# General Description of Model (cont'd)

---

- Rule-based model
- Accounting drives the simulation
- Requests for allocation are determined at beginning of run or at each timestep, depending on method selections on the accounts

# General Description of Model (cont'd)

---

- At each timestep, a **predefined rule function** solves for the supplies that represent allocations from “natural flow” to prioritized rights
- Other rules implement policy decisions (e.g., releases from reservoirs to meet unmet allocation needs)
- Rules copy slot values (diversions, transfers, releases) from accounting world to physical world to drive the physical simulation

# Predefined Rule Function

---

- **SolveWaterRights** is called in a rule:
  - SolveWaterRights(  
    subbasin-name,  
    water-type,  
    range of dates for prioritized accounts)
  - Return value is a list of {slot name,value} pairs for:
    - Supplies that represent appropriations
    - Allocation Requests, Initial Requests
    - Current Maximum Call for instream flow accounts
  - Calling rule puts the returned values into their associated slots

# Predefined Rule Function (cont'd)

---

- Current timesteps' values are copied to slots in cloned accounts
- Solver computes solution for current timestep
- Three passes:
  - Pass 1: clear all allocations
  - Pass 2: execute **solution algorithm**
  - Pass 3: transfer results as (slot, value) pairs to the calling rule

# Solution Algorithm

---

- For each right in priority order:
- Compute its allocation request (taking into account state-of-system constraints)
- Propose the maximum allocation possible (make the allocation on the cloned account), up to the requested amount but not exceeding the right's shortage or the water available at point of allocation
- Re-solve the cloned accounting system
- Check for shortages at downstream seniors
  - If a shortage occurs make a "call": using that shortage, solve backwards for the amount we have to reduce the proposed allocation
  - Take into account loss (and lag, eventually) between the shortage point and the allocation point
  - If the proposed allocation is cut back, re-solve the cloned accounting system
- Continue with next right in priority



# Solution Algorithm (cont'd)

## Special Handling for Instream Flow Rights

---

- To model instream flow rights, we need to
  - establish the natural flows at control points at the time of allocation (*Current Maximum Call* slot values)
  - execute rules to operate the reservoirs, possibly releasing water to satisfy contracts with downstream diverters
  - allocate water from the natural flows but with instream flow rights now considering natural flows and project water

# Solution Algorithm (cont'd)

## Special Handling for Instream Flow Rights

---

- Variant of the rule function, **SolveWaterRightsWithoutInstreamFlowCalls()** treats instream flow accounts differently
  - When visiting instream flow accounts (in priority order), computes their allocation requests and sets the *Current Maximum Call* to lesser of (appropriation request, current flow)
  - Does not let instream flow rights place calls
- **SolveWaterRights()** allows instream flow accounts to make calls, treating *Current Maximum Call* as an upper bound

# Status of Work

---

## ➤ Coded, in Beta test

- New water account type - instream flow account
- Dependent categories and slots on accounts
- Methods and slots for accounts representing water rights
  - Priority dates
  - Requests, shortages - slots & methods on all legal water account types
  - Disaggregation of aggregated annual requests into daily/monthly requests for diversion rights
  - Minimum bypass constraints (legal) on diversion and storage rights
  - Reference level-based requests for instream flow rights
- New solution methods on diversion accounts
  - Fractional return flow, variable fractional return flow

# Status of Work (cont'd)

---

## ➤ Coded, in Beta test, cont'd

- Methods and slots on physical objects to support water rights
  - Inflow needed to fill conservation pool, considering evap/precip
  - Capacity of physical diversion structure on reaches and reservoirs
- Rule functions to allocate water
  - Cloning of accounts to provide side-effect-free computations
  - Topological analysis of subbasin to facilitate solution, performance
  - Variants to treat instream flow rights as required

## ➤ In Progress

- On-line help
- Continued testing

# Status of Work

---

## ➤ Areas for Future Work

- Lags
- New controller (solves accounting first, followed by physical simulation)
  - Methods to move hydrologic inflow from simulation objects to accounts
  - Revisit reconciliation
- Additional legal constraints (e.g., accrual-based)
- Accrual methods to consider a subset of the supplies (e.g., natural flow only)
- Offstream reservoir storage requests to include diversion demands on the reservoir at current timestep
- New configuration possibilities for accounts on agg diversion sites
- Improved diagnostics, detection of modeling errors
- Additional performance analysis and improvements

# Summary

---

- Models priority-based rights (“first in time, first in right”)
- Flexible mechanisms for modeling various legal entitlements and constraints
- Enforces physical constraints
- Simple calls to rule functions to solve for allocations
- Performance improvement realized
- Allows RiverWare model to replicate WAM
- Much is finished, more to do

# Feedback

---

- Questions/Suggestions?