# Appendix H: Individual LRG Rules

The individual rules in the URGWOM ruleset are discussed separately and are grouped based on policy group. Each section in this appendix is associated with a different LRG policy group. The rules fire in reverse order within URGWOM, so the discussion of the policy groups begins with the last policy group associated with LRG operations, and proceeds to the first.

The discussion for each rule includes a description of the rule including an explanation of the Rule Logic, a list of slots in the model associated with the rule, a log of when and how the rule was last modified, a view of the actual RiverWare rule code, a list of Execution Constraints, and a list of functions referenced in the rule.  These items are stored within the ruleset itself and this appendix is generated automatically from that ruleset so that updates to rule documentation can be easily added whenever a rule is updated.

Model information for the URGWOM version from which this Appendix was generated, including when the model was last saved, by whom, in what version of RiverWare, and the Object Count is as follows:

## Control Display Icon1 RPL Object IconLRG Allocations

### Control Display Icon1.1 RPL Object IconInitialize Diversion Objects

Rule Purpose:   
This rule sets the Diversion Request on every Diversion Object in URGWOM, besides LeasburgToEastside, to 0 acre-feet/day if a Diversion Request hasn't already been computed, so that the system has enough data to solve initially. In the daily model, if it is the first timestep, the assignment is made from the initial timestep through the Local Timestep, to ensure that there are enough data to solve.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule fires on the first timestep of the simulation if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. object.Diversion Request, where object is every Diversion Object in the model besides LeasburgtoEastside.

List of key model objects with slots read by the rule or child functions:  
None

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**LocalTimestep
* **RPL Object Icon**GetTimestep
* **RPL Object Icon**RunStartDate
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.2 RPL Object IconSet Gain Loss for Project Water Users

Rule Purpose:   
This rule sets the gain loss on all the Project storage accounts on Caballo Reservoir equal to zero, so that the accounts can solve initially.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule contains a FOR DO statement which is used to create a list of all the storage accounts on the Caballo reservoir object with the ProjectWaterUser WaterType using the predefined AccountNamesByWaterType function. The value for the Gain Loss series slot for every account is set equal to 0 acre-feet.   
This rule fires every timestep.

Model slots written by rule:  
1. Cabally^account.Gain Loss where account is all the the ProjectWaterUser accounts on Caballo.

List of key model objects with slots read by the rule or child functions:  
None

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

**Statements**

Referenced Functions

* **RPL Object Icon**AccountNamesByWaterType

### Control Display Icon1.3 RPL Object IconInitialize Split Ag Pumping between Deep and Shallow Aquifers

Rule Purpose:   
For every groundwater object in the Mesilla and Leasburg areas, for every timestep from the run start date through the run start date + the lag to the groundwater object, this rule sets the groundwater pumping from the shallow aquifer to 0 acre-feet/day, so that the system can solve initially.

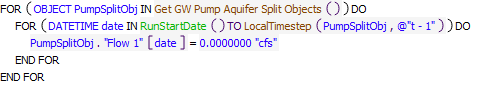
Rule Logic: Execution Constraint logic is at end of explanation.  
For every pipe junction object in URGWOM, for every timestep from the run start date through the run start date + the lag to the pipe junction object, this rule sets the value for the Flow1 series slot on the pipe junction object equal to 0 acre-feet/day.  
This rule fires on the first timestep of the simulation if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. PumpSplitObj.Flow1 where PumpSplitOpj is all 28 PipeJunction objects in the Leasburg and Mesilla regions which split groundwater pumping between the deep and shallow aquifer layers, and Flow1 is the flow coming from the shallow aquifer layer.

List of key model objects with slots read by the rule or child functions:  
None

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**Get GW Pump Aquifer Split Objects
* **RPL Object Icon**LocalTimestep
* **RPL Object Icon**GetTimestep
* **RPL Object Icon**RunStartDate
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.4 RPL Object IconInitialize EPCWID ACE Conservation Credit

Rule Purpose:   
This rule sets the EPCWID ACE Conservation Credit zero for all timesteps. The user should input these slot values if they are not zero.

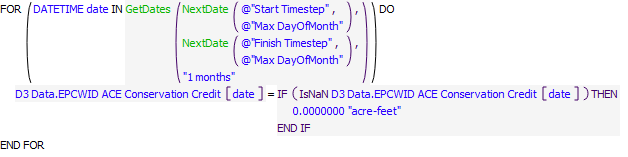
Rule Logic: Execution Constraint logic is at end of explanation.  
This rule contains a FOR DO loop to cycle through every timestep of the run. For each timestep of the run, the value for the EPCWID ACE Conservation Credit series slot on the D3 Data object is set equal to 0 acre-feet if the value has not been input, as specified by the IsNaN predefined function.   
This rule fires on the first timestep of the simulation if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.EPCWID ACE Conservation Credit

List of key model objects with slots read by the rule or child functions:  
None

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**NextDate
* **RPL Object Icon**GetDates
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.5 RPL Object IconTotal Usable Water Available for Release

Rule Purpose:   
This rule computes the Total Usable Water Available for Release, which provides an estimate of the total available project water based on project water in storage in Elephant Butte and Caballo, plus Caballo releases from February 1st of the current year to the current date, minus a minimum project pool of 5,000 acre-feet.

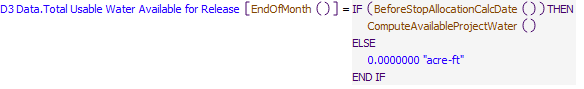
Rule Logic: Execution Constraint logic is at end of explanation.  
The end-of-month value for the Total Usable Water Available for Release series slot on the D3 Data object is set with an IF THEN ELSE statement, which checks if the current timestep is before October, with reference to the user-defined BeforeStopAllocationCalcDate function.   
If FALSE, the Total Usable Water Available for Release gets set to 0 acre-ft.   
If TRUE, the Total Usable Water Available for Release gets set equal to the ComputeAvailableProjectWater function, which is equal to the Caballo releases from February 1st of the current year through the current timestep, plus the current storage in the Rio Grande accounts on Caballo and Elephant Butte reservoirs, minus the user-input value in the MinRGPool slot on the Rio Grande Project Data object.  
This rule fires at the start of every month if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.Total Usable Water Available for Release

List of key model objects with slots read by the rule or child functions:  
1. Caballo  
2. ElephantButte  
3. Rio Grande Project Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**ComputeAvailableProjectWater
* **RPL Object Icon**BeforeStopAllocationCalcDate
* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.6 RPL Object IconSet Initial Diversion Ratio

Rule Purpose:   
This rule sets the Diversion Ratio to the Diversion Ratio from the previous month, unless the Diversion Ratio from the previous month hasn't been computed, in which case this rule sets the Diversion Ratio empirically as a function of the available project water and the maximum annual allotment for the Rio Grande Project. This initial estimate is needed to "seed" some of the other rules.

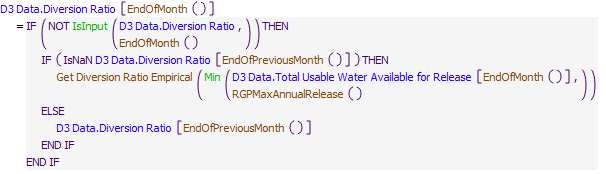
Rule Logic: Execution Constraint logic is at end of explanation.  
If the end-of-month value for the Diversion Ratio series slot on the D3 Data object is not Input, then the slot is set with an IF THEN statement, which checks if the Diversion Ratio value from the previous month is non-existent.   
If TRUE, the Diversion Ratio gets set equal to the value returned by the user-defined Get Diversion Ratio Empirical function, given that the total water released is the minimum of the values in the D3 Data.Total Usable Water Available for Release and D3 Data.RGPMaxAnnualRelease slots. The empirical diversion ratio is computed using a regression equation reported in the original 2008 OA allocation spreadsheet.   
If FALSE, the Diversion Ratio gets set equal to Diversion Ratio value from the previous month.  
This rule fires at the start of every month if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.Diversion Ratio

List of key model objects with slots read by the rule or child functions:  
1. D3 Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**Get Diversion Ratio Empirical
* **RPL Object Icon**EndOfPreviousMonth
* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**Min
* **RPL Object Icon**IsInput
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**RGPMaxAnnualRelease

### Control Display Icon1.7 RPL Object IconEstimated EOY Release for Diversion Ratio

Rule Purpose:  
This rule sets the Estimated EOY Release for Diversion Ratio equal to the Current Usable Water, unless the month is November, in which case it is set to the sum of the February through October Caballo outflows.

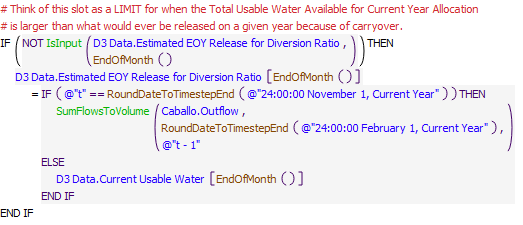
Rule Logic:  
If not already input, the end-of-month value for the Estimated EOY Release for Diversion Ratio series slot on the D3 Data object is set with an IF THEN ELSE statement, which checks if the current timestep is in November.  
If TRUE, the Estimated EOY Release for Diversion Ratio gets set to the sum of the Caballo Outflow values from February 1st of the current year through October 31st, converted to a volume.  
If FALSE, the Estimated EOY Release for Diversion Ratio gets set to the end-of-month value of Current Usable Water on the D3 Data object.   
This rule fires at the start of every month if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.Estimated EOY Release for Diversion Ratio

List of key model objects with slots read by the rule or child functions:  
1. D3 Data  
2. Caballo

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**RoundDateToTimestepEnd
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**SumFlowsToVolume
* **RPL Object Icon**IsInput
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.8 RPL Object IconSet Distric Carryover

Rule Purpose:   
If the carryover each district has at the beginning of the current year is not input, then carryover is set to the minimum of 60% of a full allocation, or the water remaining in the account from the previous year plus the any carryover excess above 60% of the full allocation of the other district. At the beginning of the first year of the simulation, the carryover for each district is set equal to the initial storage for that district.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule has four assignments.   
In the first assignment, the value for the Carry Over series slot on the EBID account on the Caballo reservoir object is set with an IF THEN ELSE statement which check if the Current Year is equal to the Start Year of the run.   
If TRUE, the Carry Over is set equal to the value in the Storage series slot on the EBID account on the Caballo reservoir object.   
If FALSE, another IF THEN ELSE statements checks if the EBID Carryover from Previous Year value on the D3 Data object is input.   
IF TRUE, then the Carry Over is set equal to this input value.  
IF FALSE, then Carry Over is set equal to the minimum of 0.6 times the value in the EBID column in the Full District Allocations tableslot on the D3 Data object, and the value in the Storage series slot on the EBID account on the Caballo reservoir object for the previous timestep plus the max of 0 acre-feet and the value for the Storage series slot on the EPCWID account on the Caballo reservoir object for the previous timestep minus 0.6 times the value in the first row and the column corresponding to EPCWID in the Full District Allocations tableslot on the D3 Data object.   
In the second assignment, the value for the Carry Over series slot on the EPCWID account on the Caballo reservoir object is set in the same way that the EBID carryover was set.  
In the third assignment, the EBID Carryover from Previous Year value on the D3 Data object is set equal to 0 acre-feet if it is not already Input.   
In the fourth assignment, the EPCWID Carryover from Previous Year value on the D3 Data object is set equal to 0 acre-feet if it is not already Input.

This rule fires on the first timestep of each year if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. Caballo^EBID.Carry Over  
2. Caballo^EPCWID.Carry Ove  
3. D3 Data.EBID Carryover from Previous Year  
4. D3 Data.EPCWID Carryover from Previous Year

List of key model objects with slots read by the rule or child functions:  
1. D3 Data  
2. Caballo

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**EndOfCurrentYear
* **RPL Object Icon**IsFirstTimestepOfYear
* **RPL Object Icon**Min
* **RPL Object Icon**Max
* **RPL Object Icon**IsInput
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.9 RPL Object IconEstimated EOY Carryover

Rule Purpose:  
This rule sets the estimated end-of-year carryover for the two districts to 0 acre-feet unless these values are input.

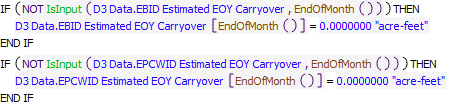
Rule Logic:  
This rule sets the end-of-month value for the EBID Estimated EOY Carryover slot on the D3 Data object to 0 acre-feet unless a value is already input. The rule does the same thing for EPCWID.  
This rule fires at the start of every month if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.EBID Estimated EOY Carryover  
2. D3 Data.EPCWID Estimated EOY Carryover

List of key model objects with slots read by the rule or child functions:  
None

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**IsInput
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.10 RPL Object IconRecompute Diversion Ratio

Rule Purpose:   
This rule computes the Diversion Ratio as the sum of all the river headgate diversions from February through the previous month divided by the sum of the Caballo reservoir releases from February through the previous month, unless Caballo has spilled this year, in which case the Diversion Ratio is computed empirically.

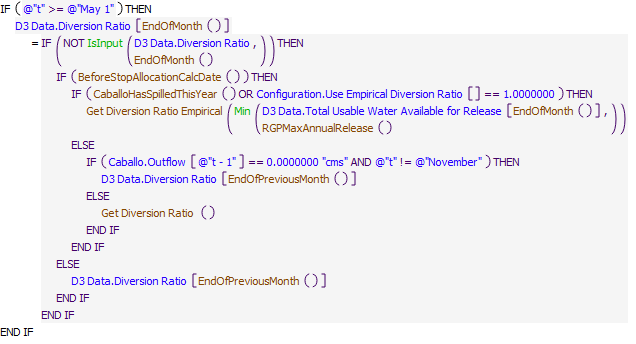
Rule Logic: Execution Constraint logic is at end of explanation.  
If the end-of-month value for the Diversion Ratio series slot on the D3 Data object is not Input, then the slot is set with an IF THEN statement, which checks if the current timestep is before October, with reference to the user-defined BeforeStopAllocationCalcDate function.   
If FALSE, the Diversion Ratio gets set equal to Diversion Ratio value from the previous month.   
If TRUE, another IF THEN ELSE statement checks if Caballo has spilled this year, or if the Use Empirical Diversion Ratio switch on the Configuration object is set to 1.   
If TRUE, the Diversion Ratio gets set equal to the value returned by the user-defined Get Diversion Ratio Empirical function, given that the total water released is the minimum of the values in the D3 Data.Total Usable Water Available for Release and D3 Data.RGPMaxAnnualRelease slots. The empirical diversion ratio is computed using a regression equation reported in the original 2008 OA allocation spreadsheet.   
If FALSE, another IF THEN ELSE statement checks if Caballo outflow is zero and if the month is not November.  
If TRUE, the Diversion Ratio gets set equal to Diversion Ratio value from the previous month.   
If FALSE, the Diversion Ratio gets set equal to the value returned by the user-defined Get Diversion Ratio function, which divides the sum of the year-to-date diversions at all of the LRG river headgates by the sum of the year-to-date Caballo releases.   
This rule fires at the start of every month if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.Diversion Ratio

List of key model objects with slots read by the rule or child functions:  
1. Configuration  
2. Caballo  
3. D3 Data  
4. Compact Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**Get Diversion Ratio
* **RPL Object Icon**Get Diversion Ratio Empirical
* **RPL Object Icon**BeforeStopAllocationCalcDate
* **RPL Object Icon**CaballoHasSpilledThisYear
* **RPL Object Icon**EndOfPreviousMonth
* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**Min
* **RPL Object Icon**IsInput
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**RGPMaxAnnualRelease

### Control Display Icon1.11 RPL Object IconTotal Usable Water Available for Current Year Allocation

Rule Purpose:   
This rule computes the Total Usable Water Available for Current Year Allocation, which is the total available project water minus the district carryover adjusted by the diversion ratio.

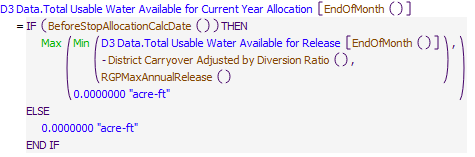
Rule Logic: Execution Constraint logic is at end of explanation.  
The end-of-month value for the Total Usable Water Available for Current Year Allocation series slot on the D3 Data object is set with an IF THEN ELSE statement, which checks if the current timestep is before October, with reference to the user-defined BeforeStopAllocationCalcDate function.   
If FALSE, the Total Usable Water Available for Current Year Allocation gets set to 0 acre-ft.  
If TRUE, the Total Usable Water Available for Current Year Allocation gets set to the minimum of the end-of-month value on the Total Usable Water Available for Release slot on the D3 Data object minus the District Carryover Adjusted by Diversion Rate user-defined function, or the RGPMaxAnnualReleaseuser-defined function, constrained to be no less than 0 acre-feet.  
This rule fires at the start of every month if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.Total Usable Water Available for Current Year Allocation

List of key model objects with slots read by the rule or child functions:  
1. D3 Data  
2. Caballo

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**District Carryover Adjusted by Diversion Ratio
* **RPL Object Icon**BeforeStopAllocationCalcDate
* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**Min
* **RPL Object Icon**Max
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**RGPMaxAnnualRelease

### Control Display Icon1.12 RPL Object IconCompute Current Usable Water

Rule Purpose:   
This rule computes the Current Usable Water, which is the amount anticipated for release and includes a projected amount of carryover water use plus new project water use.

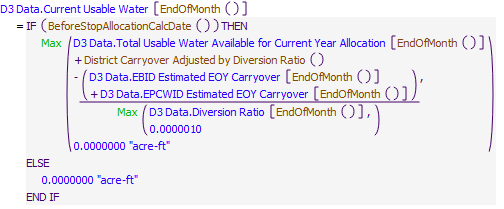
Rule Logic: Execution Constraint logic is at end of explanation.  
The end-of-month value for the Current Usable Water series slot on the D3 Data object is set with an IF THEN ELSE statement, which checks if the current timestep is before October, with reference to the user-defined BeforeStopAllocationCalcDate function.   
If FALSE, the Current Usable Water gets set to 0 acre-ft.  
If TRUE, the Current Usable Water gets set to the end-of-month value on the Total Usable Water Available for Current Year Allocation slot on the D3 data object plus the value returned by the user-defined District Carryover Adjusted by Diversion Ratio function minus the sum of the end-of-month values in the EBID Estimated EOY Carryover and EPCWID EOY Carryover series slots on the D3 Data object divided by the value in the Diversion Ratio series slot on the D3 Data object, limited to be no less than 0 acre-feet.   
This rule fires at the start of every month if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.Current Usable Water

List of key model objects with slots read by the rule or child functions:  
1. D3 Data  
2. Caballo

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**District Carryover Adjusted by Diversion Ratio
* **RPL Object Icon**BeforeStopAllocationCalcDate
* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**Max
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.13 RPL Object IconSet D2 Drought Correction Factor

Rule Purpose:  
If enabled, this rule computes the D2 Drought Correction Factor (identified in the RGP allocation spreadsheet as Multiyear Extreme Drought D2 Correction Factor), using the formulation described in section 4.10 of Appendix B of the 2012 RGP Operations Manual. If the Caballo outflow volume the previous year was more than 400,000 acre-feet, then the Drought Correction Factor is 1. If the Caballo outflow volume the previous year was less than 400,000 acre-feet, then the Drought Correction Factor is 0.88. If the Caballo outflow volume in each of the two previous years was less than 400,000 acre-feet each year, then the Drought Correction Factor is 0.78. If the Caballo outflow volume in each of the three previous years was less than 400,000 acre-feet each year, then the Drought Correction Factor is 0.75. This Drought Correction Factor is used in the computation of the Gross D2 Diversion Allocation.

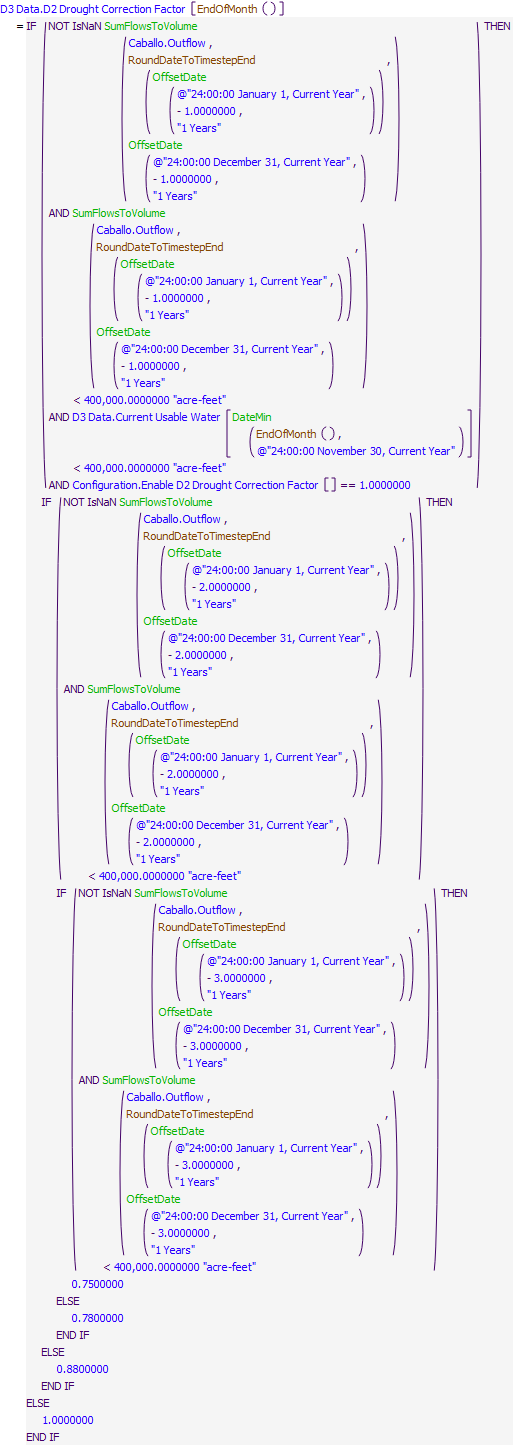
Rule Logic:  
If the Enable D2 Drought Correction Factor switch on the Configuration object is set to 1, then this rule sets the end-of-month value on the D2 Drought Correction Factor slot on the D3 Data object equal with an IF THEN ELSE statement, which checks if there are data for the previous year's Caballo release volume, and if the previous year's Caballo release volume is less than 400,000 acre-feet.  
IF FALSE, then the D2 Drought Correction Factor is set to 1.  
IF TRUE, then another IF THEN ELSE statement checks if there are data for Caballo's release volume two years prior, and if the Caballo release volume two years prior is less than 400,000 acre-feet.  
IF FALSE, then the D2 Drought Correction Factor is set to 0.88.  
IF TRUE, then another IF THEN ELSE statement checks if there are data for Caballo's release volume three years prior, and if the Caballo release volume three years prior is less than 400,000 acre-feet.  
IF FALSE, then the D2 Drought Correction Factor is set to 0.78.  
IF TRUE, then the D2 Drought Correction Factor is set to 0.75.  
This rule fires at the start of every month if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.D2 Drought Correction Factor

List of key model objects with slots read by the rule or child functions:  
1. Configuration  
2. Caballo  
3. D3 Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**RoundDateToTimestepEnd
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**SumFlowsToVolume
* **RPL Object Icon**DateMin
* **RPL Object Icon**OffsetDate
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.14 RPL Object IconCompute D1 and D2 Allocations

Rule Purpose:   
This rule computes the D1 and D2 Allocations based on the D1 and D2 curves. The D1 Allocation is based on the Current Usable Water and the Estimated EOY Release for Diversion Ratio, and the D2 Gross Allocation is based on the Total Usable Water Available for Current Year Allocation and the D2 Drought Correction Factor.

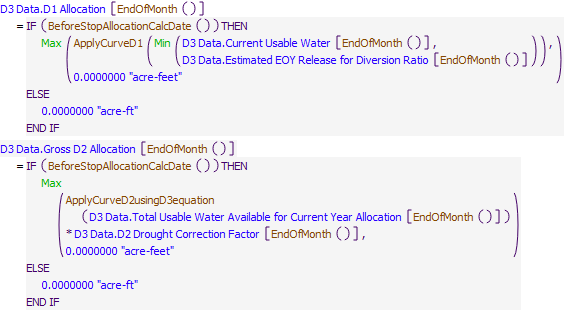
Rule Logic: Execution Constraint logic is at end of explanation.  
This rule has two assignments. In the first assignment, the end-of-month value for the D1 Allocation series slot on the D3 Data object is set with an IF THEN ELSE statement, which checks if the current timestep is before October, with reference to the user-defined BeforeStopAllocationCalcDate function.  
If FALSE, the D1 Allocation gets set to 0 acre-feet.   
If TRUE, the D1 Allocation gets set to the maximum of 0 acre-feet, or the value returned by the user-defined ApplyCurveD1function based on the minimum of the Current Usable Water and Estimated EOY Release for Diversion Ratio values from the D3 Data object.   
In the second assignment, the end-of-month value for the Gross D2 Allocation series slot on the D3 Data object is set with an IF THEN ELSE statement, which checks if the current timestep is within the irrigation season, which is February through October, with reference to the user-defined IsIrrigationSeason function.   
If FALSE, the Gross D2 Allocation gets set to 0 acre-feet.   
If TRUE, the Gross D2 Allocation gets set to the maximum of 0 acre-feet, or the value returned by the user-defined ApplyCurveD2usingD3equation function based on the Total Usable Water Available for Current Year Allocation value multiplied by the D2 Drought Correction Factor, from the D3 Data object.  
This rule fires at the start of every month if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.D1 Allocation  
2. D3 Data.Gross D2 Allocation

List of key model objects with slots read by the rule or child functions:  
1. D3 Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**ApplyCurveD1
* **RPL Object Icon**BeforeStopAllocationCalcDate
* **RPL Object Icon**ApplyCurveD2usingD3equation
* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**Min
* **RPL Object Icon**Max
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.15 RPL Object IconSet Mexico Allocation

Rule Purpose:   
The rule sets the Mexico Allocation to 11.34% of the D1 Allocation, limited to the maximum allocation of 60,000 acre-feet.

Rule Logic:  
The end-of-month value for the Mexico Allocation series slot the D3 Data object gets set to the minimum of the value on the D1 Allocation series slot on the D3 Data object multiplied by the value on the Percent of D1 Allocation for Mexico scalar slot on the D3 Data object, or the value on the Max Mexico Allocation scalar slot on the D3 Data object.   
The rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.Mexico Allocation

List of key model objects with slots read by the rule or child functions:  
1. D3 Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**Min
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.16 RPL Object IconSet Mexico Account Slot Inflow

Rule Purpose:   
This rule sets the "slot inflow" slot on Mexico's account on Caballo reservoir. In the first allotment month (February), the slot inflow gets the full amount of the allocation as computed. In subsequent months, the slot inflow is set to the incremental additional allocation over and above that computed for the previous month(s).

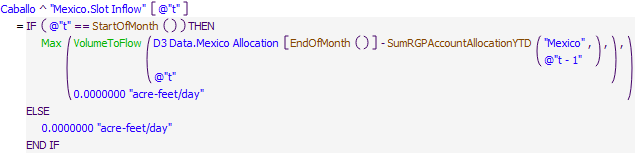
Rule Logic: Execution Constraint logic is at end of explanation.  
The value for the Slot Inflow series slot on the Mexico account on the Caballo reservoir object is set with an IF THEN ELSE statement, which checks if it is the start of the month.  
If TRUE, the Slot Inflow gets set to 0 acre-feet/month.   
If FALSE, the Slot Inflow gets set to the maximum of 0 acre-feet or the Mexico Allocation on the D3 Data object minus the value returned by the user-defined SumRGPAccountAllocationYTD function for the Mexico account.   
The rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. Caballo^Mexico.Slot Inflow

List of key model objects with slots read by the rule or child functions:  
1. D3 Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**SumRGPAccountAllocationYTD
* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**Max
* **RPL Object Icon**VolumeToFlow
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.17 RPL Object IconSet Initial District Allocations

Rule Purpose:   
This rule makes two assignments. It sets the EPCWID D2 allocation to 43.226% of the net D2 allocation and it sets the EBID D2 allocation to 56.774% of the net D2 allocation.

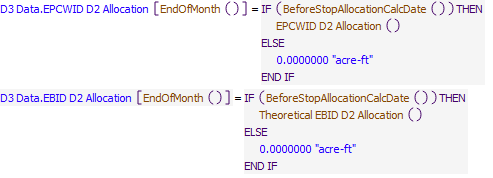
Rule Logic: Execution Constraint logic is at end of explanation.  
This rule makes two assignments. In the first assignment, the end-of-month value for the EPCWID D2 Allocation series slot on the D3 Data object is set with an IF THEN ELSE statement, which checks if the current timestep is before October, with reference to the user-defined BeforeStopAllocationCalcDate function.  
IF FALSE, the EPCWID D2 Allocation is set to 0 acre-ft.  
IF TRUE, the EPCWID D2 Allocation is set equal to the value returned by the user-defined EPCWID D2 Allocation function. The EPCWID D2 Allocation function is the Net D2 Allocation for EBID and EPCWID (the Gross D2 Allocation minus the Mexico Allocation) multiplied by the value in the EPCWID column of the AllocationPctbyDistrict tableslot on the Rio Grande Project Data object, constrained to be no less than 0 acre-ft.  
In the second assignment, the end-of-month value for the EBID D2 Allocation series slot on the D3 Data object is set with an IF THEN ELSE statement, which checks if the current timestep is before October, with reference to the user-defined BeforeStopAllocationCalcDate function.  
IF FALSE, the EBID D2 Allocation is set to 0 acre-ft.  
IF TRUE, the EBID D2 Allocation is set equal to the value returned by the user-defined Theoretical EBID D2 Allocation function. The Theoretical EBID D2 Allocation function is the Net D2 Allocation for EBID and EPCWID (the Gross D2 Allocation minus the Mexico Allocation) multiplied by the value in the EBID column of the AllocationPctbyDistrict tableslot on the Rio Grande Project Data object, constrained to be no less than 0 acre-ft.  
This rule fires at the start of every month if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.EPCWID D2 Allocation  
2. D3 Data.EBID D2 Allocation

List of key model objects with slots read by the rule or child functions:  
1. D3 Data  
2. Rio Grande Project Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**EPCWID D2 Allocation
* **RPL Object Icon**BeforeStopAllocationCalcDate
* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**Theoretical EBID D2 Allocation

### Control Display Icon1.18 RPL Object IconSet Diversion Ratio Adjustment

Rule Purpose:   
This rule sets the Diversion Ratio Adjustment, which is computed as the Current Usable Water multiplied by (the diversion ratio minus 1). It is used to adjust the EBID allotment to account for variability in delivery efficiency.

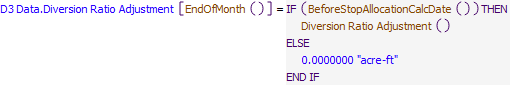
Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the end-of-month value for the Diversion Ratio Adjustment series slot on the D3 Data object with an IF THEN ELSE statement, which checks if the current timestep is before October, with reference to the user-defined BeforeStopAllocationCalcDate function.  
IF FALSE, the Diversion Ratio Adjustment is set to 0 acre-ft.  
IF TRUE, the Diversion Ratio Adjustment is set equal to the value returned by the user-defined Diversion Ratio Adjustment function. The Diversion Ratio Adjustment function is the Current Usable Water multiplied by the Diversion Ratio minus 1.  
This rule fires at the start of every month if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.Diversion Ratio Adjustment

List of key model objects with slots read by the rule or child functions:  
1. D3 Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**Diversion Ratio Adjustment
* **RPL Object Icon**BeforeStopAllocationCalcDate
* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.19 RPL Object IconSet EBID Diversion Ratio Allocation

Rule Purpose:   
This rule sets the EBID diversion ratio allocation.

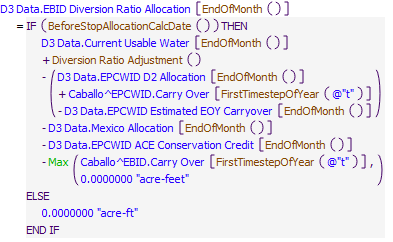
Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the end-of-month value for the EBID Diversion Ratio Allocation series slot on the D3 Data object with an IF THEN ELSE statement, which checks if the current timestep is before or equal to October, with reference to the user-defined BeforeStopAllocationCalcDate function.   
If FALSE, the value for the EBID Diversion Ratio Allocation gets set to 0 acre-ft.  
If TRUE, the value for the EBID Diversion Ratio Allocation gets set equal to the Current Usable Water plus the value returned by the Diversion Ratio Adjustment function (which is the Current Usable Water multiplied by the Diversion Ratio minus 1) minus the EPCWID D2 Allocation minus the EPCWID Carryover from the previous year minus the EPCWID Estimated Carryover for the current year minus the Mexico Allocation minus the EPCWID ACE Conservation Credit minus the maximum of 0 acre-feet or the EBID Carry Over from the previous year.  
This rule fires at the start of every month if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.EBID Diversion Ratio Allocation

List of key model objects with slots read by the rule or child functions:  
1. D3 Data  
2. Caballo

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**Diversion Ratio Adjustment
* **RPL Object Icon**BeforeStopAllocationCalcDate
* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**FirstTimestepOfYear
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**Max
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.20 RPL Object IconSet EBID Diversion Allocation

Rule Purpose:   
This rule sets the EBID Diversion Allocation.

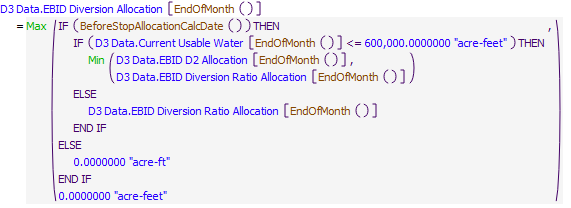
Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the end-of-month value for the EBID Diversion Allocation series slot on the D3 Data object with an IF THEN ELSE statement, which checks if the current timestep is before or equal to October, with reference to the user-defined BeforeStopAllocationCalcDate function.   
If FALSE, the value for the EBID Diversion Allocation gets set to 0 acre-ft.  
If TRUE, an internal IF THEN ELSE statement checks if the Current Usable Water is less than or equal to 600,000 acre-feet.   
If TRUE, the EBID Diversion Allocation gets set equal to the minimum of the EBID D2 Allocation or the EBID Diversion Ratio Allocation.  
If FALSE, the EBID Diversion Allocation gets set equal to the EBID Diversion Ratio Allocation.  
The EBID Diversion Allocation is limited to be no less than 0 acre-feet.   
This rule fires at the start of every month if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.EBID Diversion Allocation

List of key model objects with slots read by the rule or child functions:  
1. D3 Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**BeforeStopAllocationCalcDate
* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**Min
* **RPL Object Icon**Max
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.21 RPL Object IconSet Final EBID Diversion Allocation

Rule Purpose:   
This rule sets the EBID Final Diversion Allocation.

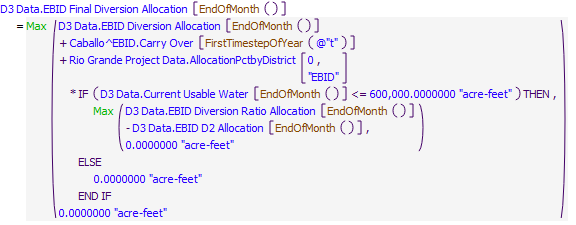
Rule Logic: Execution Constraint logic is at end of explanation.  
The end-of-month value for the EBID Final Diversion Allocation series slot on the D3 Data object is set equal to the EBID Diversion Allocation plus the EBID Carry Over plus the value in the EBID column in the AllocationPctbyDistrict tableslot on the Rio Grande Project Data object multiplied by the value returned by an IF THEN ELSE statement, which checks if the Current Usable Water is less than 600,000 acre-feet. If FALSE, the value returned is 0 acre-feet. If TRUE, the value returned in the maximum of 0 acre-feet or the EBID Diversion Ratio Allocation minus the EBID D2 Allocation.  
The EBID Final Diversion Allocation is limited to be no less than 0 acre-feet.   
This rule fires at the start of every month if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.EBID Final Diversion Allocation

List of key model objects with slots read by the rule or child functions:  
1. D3 Data  
2. Rio Grande Project Data  
3. Caballo

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**FirstTimestepOfYear
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**Max
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.22 RPL Object IconSet Final EPCWID Diversion Allocation

Rule Purpose:   
This rule sets the EPCWID Final Diversion Allocation.

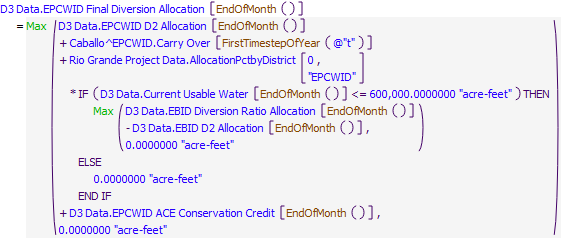
Rule Logic: Execution Constraint logic is at end of explanation.  
The end-of-month value for the EPCWID Final Diversion Allocation series slot on the D3 Data object is set equal to the EPCWID D2 Allocation plus the EPCWID Carry Over plus EPCWID ACE Conservation Credit plus the value in the EPCWID column in the AllocationPctbyDistrict tableslot on the Rio Grande Project Data object multiplied by the value returned by an IF THEN ELSE statement, which checks if the Current Usable Water is less than 600,000 acre-feet. If FALSE, the value returned is 0 acre-feet. If TRUE, the value returned in the maximum of 0 acre-feet or the EBID Diversion Ratio Allocation minus the EBID D2 Allocation.  
The EPCWID Final Diversion Allocation is limited to be no less than 0 acre-feet.   
This rule fires at the start of every month if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. D3 Data.EPCWID Final Diversion Allocation

List of key model objects with slots read by the rule or child functions:  
1. D3 Data  
2. Rio Grande Project Data  
3. Caballo

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**FirstTimestepOfYear
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**Max
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.23 RPL Object IconSet Total Allocations

Rule Purpose:   
This rule sets the Total Allocation equal to the sum of the Mexico Allocation, the EBID Final Diversion Allocation and the EPCWID Final Diversion Allocation.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the end-of-month value for the Total Allocation series slot on the D3 Data object equal to the sum of the values in the Mexico Allocation, EBID Final Diversion Allocation, and EPCWID Final Diversion Allocation series slots on the D3 Data object.   
This rule fires at the start of every month.

Model slots written by rule:  
1. D3 Data.Total Allocation

List of key model objects with slots read by the rule or child functions:  
1. D3 Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**StartOfMonth

### Control Display Icon1.24 RPL Object IconSet Allocation on EPCWID Account

Rule Purpose:   
This rule sets the "slot inflow" slot on EPCWID's account. In the first allocation month (January), the slot inflow gets the full amount of the allocation as computed (EPCWID Final Diversion Allocation; no carryover is included as it is already in the account). In subsequent months, the slot inflow is set to the incremental additional allocation over and above that computed for the previous month(s).

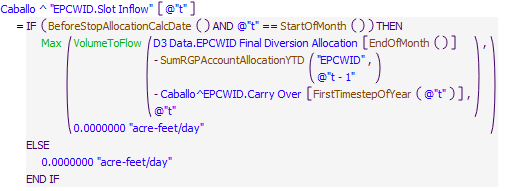
Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the value for the Slot Inflow series slot on the EPCWID account on the Caballo reservoir object is set with an IF THEN ELSE statement, which checks if the current timestep is before October, with reference to the user-defined BeforeStopAllocationCalcDate function, and if it is the first timestep of the month.   
If FALSE, the Slot Inflow gets set to 0 acre-feet/month.   
If TRUE, the Slot Inflow gets set to EPCWID Final Diversion Allocation minus the value returned by the SumRGPAccountAllocationYTD function for the EPCWID account, minus the EPCWID Carryover from the previous year.   
This rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. Caballo^EPCWID.Slot Inflow

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Caballo  
3. CaballoData  
4. ElephantButte  
5. D3 Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**SumRGPAccountAllocationYTD
* **RPL Object Icon**BeforeStopAllocationCalcDate
* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**FirstTimestepOfYear
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**Max
* **RPL Object Icon**VolumeToFlow
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.25 RPL Object IconSet WetDryAvg

Rule Purpose:   
This rule sets the WetDryAverageIndex each month of the irrigation season depending on the total project usable water. Total usable water less than 435,000 acre-feet is considered a dry year. Greater than 435,000 but less than 700,000 is considered average, and greater than 700,000 is considered wet. The index may be reset at any time based on changes in hydrologic conditions. The index is used to determine the appropriate release and diversion request patterns for subsequent rules.

Rule Logic: Execution Constraint logic is at end of explanation.  
The WetDryAvgIndex series slot on the Rio Grande Project Data object is set with an IF THEN ELSE statement which checks if the value returned by the ComputeAvailableProjectWater function is less than or equal to the value in the Dry column in the WetDryAvgIndexTable tableslot on the Rio Grande Project Data object.   
If TRUE, the WetDryAvgIndex gets set to 0.   
If FALSE, another internal IF THEN ELSE statement is used to checks if the value returned by the ComputeAvailableProjectWater function is less than or equal to the value in the Average column in the WetDryAvgIndexTable tableslot.  
If TRUE, the WetDryAvgIndex gets set to 1.   
If FALSE, the WetDryAvgIndex gets set to 2.   
This rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. Rio Grande Project Data.WetDryAvgIndex

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Caballo  
3. Rio Grande Project Data  
4. ElephantButte

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**ComputeAvailableProjectWater
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.26 RPL Object IconSet Allocation on EBID Account

Rule Purpose:   
This rule sets the "slot inflow" slot on EBID's account. In the first allocation month (January), the slot inflow gets the full amount of the allocation as computed (EBID Final Diversion Allocation; no carryover is included as it is already in the account). In subsequent months, the slot inflow is set to the incremental additional allocation over and above that computed for the previous month(s)

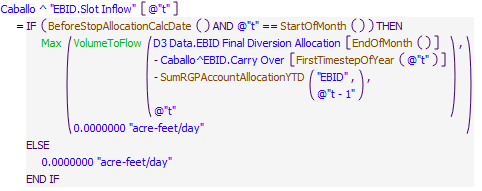
Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the value for the Slot Inflow series slot on the EBID account on the Caballo reservoir object is set with an IF THEN ELSE statement, which checks if the current timestep is before October, with reference to the user-defined BeforeStopAllocationCalcDate function, and if it is the first timestep of the month.   
If FALSE, the Slot Inflow gets set to 0 acre-feet/month.   
If TRUE, the Slot Inflow gets set to EBID Final Diversion Allocation minus the value returned by the SumRGPAccountAllocationYTD function for the EBID account, minus the EBID Carryover from the previous year.   
This rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. Caballo^EBID.Slot Inflow

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Caballo  
3. CaballoData  
4. ElephantButte  
5. D3 Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**SumRGPAccountAllocationYTD
* **RPL Object Icon**BeforeStopAllocationCalcDate
* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**FirstTimestepOfYear
* **RPL Object Icon**StartOfMonth
* **RPL Object Icon**Max
* **RPL Object Icon**VolumeToFlow
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon1.27 RPL Object IconInitialize Elephant Butte Release

Rule Purpose:   
This rule sets the outflow from Elephant Butte reservoir to zero acre-feet/day. This will allow the system to solve after the river headgate demands have been set.

Rule Logic: Execution Constraint logic is at end of explanation.  
The value for the Outflow series slot on the Elephant Butte reservoir object is set equal to 0 acre-feet/day.   
This rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function, and if the current timestep is after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay.

Model slots written by rule:  
1. ElephantButte.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon1.28 RPL Object IconInitialize Caballo Release

Rule Purpose:   
This rule sets the outflow from Caballo reservoir to zero acre-feet/day. This will allow the system to solve after the river headgate demands have been set.

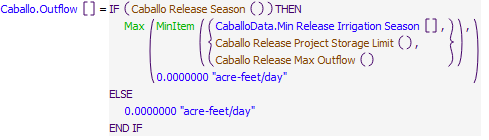
Rule Logic: Execution Constraint logic is at end of explanation.  
The value for the Outflow series slot on the Caballo reservoir object is set equal to 0 acre-feet/day.   
This rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function, if the current timestep is after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay, and if Caballo Outflow is not already input.

Model slots written by rule:  
1. Caballo.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Caballo  
3. CaballoData  
4. D3 Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**Caballo Release Season
* **RPL Object Icon**Caballo Release Project Storage Limit
* **RPL Object Icon**Caballo Release Max Outflow
* **RPL Object Icon**MinItem
* **RPL Object Icon**Max
* **RPL Object Icon**IsInput
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

## Control Display Icon2 RPL Object IconLRG Diversions

### Control Display Icon2.1 RPL Object IconSet D3 Diversion Requests Based on Diversion Pattern

Rule Purpose  
Sets the diversion requests for all Project River Head Gates based on each District's Final Allocation, demand coefficients that vary by hydrologic year type (wet, dry, or average), and coefficients which split district allocations between the various river headgates in that district.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule has six assignments, one for every river headgate in the LRG portion of the model.

If not already input, the first assignment sets the value on the Diversion Request slot on the ArreyandPercha diversion object at the RinconLocalTimestep (the Caballo timestep) with an IF THEN ELSE statement, which checks if the RinconLocalTimestep is after or equal to the date returned by the LRGDiversionStartDate function for EBID.  
If FALSE, the ArreyandPercha Diversion Request gets set to 0 acre-ft.  
If TRUE, the ArreyandPercha Diversion Request gets set to the value in the ArreyandPercha row in the AnnualRHGDistribution tableslot multiplied by the end-of-month EBID Final Diversion Allocation multiplied by the value returned by an IF THEN ELSE statement which checks if the timestep is daily. If TRUE, the value returned is the value from the RinconLocalTimestep row and the GetYearTypeasString column in the ArreyandPerchaDiversionPattern periodic tableslot. If FALSE, the value returned is the value returned by the Monthly Pattern function for the ArreyandPerchaDiversionPattern periodic tableslot (which aggregates the daily diversion pattern coefficients to monthly).  
The ArreyandPercha Diversion Request is limited to the value in the Max Diversion slot on the ArreyandPercha diversion object.

If not already input, the second assignment sets the value on the Diversion Request slot on the LeasburgandPicacho diversion object at the LeasburgLocalTimestep (the Caballo timestep + 1 day in the daily model) in the same way that the ArreyandPercha Diversion Request was set.

If not already input, the third assignment sets the value on the Diversion Request slot on the Westside diversion object at the MesillaLocalTimestep (the Caballo timestep + 2 days in the daily model) equal to the SUM of:  
1. The value returned by an IF THEN ELSE statement, which checks if the MesillaLocalTimestep is after or equal to the date returned by the LRGDiversionStartDate function for EBID.  
If FALSE, the value returned is 0 acre-feet/day.  
IF TRUE, the value returned is the value in the WestsideNM row in the AnnualRHGDistribution tableslot multiplied by the end-of-month EBID Final Diversion Allocation multiplied by the value returned by an IF THEN ELSE statement which checks if the timestep is daily. If TRUE, the value returned is the value from the MesillaLocalTimestep row and the GetYearTypeasString column in the WestsideDiversionPattern periodic tableslot. If FALSE, the value returned is the value returned by the Monthly Pattern function for the WestsideDiversionPattern periodic tableslot (which aggregates the daily diversion pattern coefficients to monthly).  
2. The value returned by an IF THEN ELSE statement, which checks if the MesillaLocalTimestep is after or equal to the date returned by the LRGDiversionStartDate function for EPCWID.  
If FALSE, the value returned is 0 acre-feet/day.  
IF TRUE, the value returned is the value in the WestsideTX row in the AnnualRHGDistribution tableslot multiplied by the end-of-month EPCWID Final Diversion Allocation multiplied by the value returned by an IF THEN ELSE statement which checks if the timestep is daily. If TRUE, the value returned is the value from the MesillaLocalTimestep row and the GetYearTypeasString column in the WestsideDiversionPattern periodic tableslot. If FALSE, the value returned is the value returned by the Monthly Pattern function for the WestsideDiversionPattern periodic tableslot (which aggregates the daily diversion pattern coefficients to monthly).  
The Westside Diversion Request is limited to the value in the Max Diversion slot on the Westside diversion object.

If not already input, the fourth assignment sets the value on the Diversion Request slot on the Eastside diversion object at the MesillaLocalTimestep (the Caballo timestep + 2 days in the daily model) in the same way that the Westside Diversion Request was set.

If not already input, the fifth assignment sets the value on the Diversion Request slot on the American Diversion object at the BlwCourchesneLocalTimestep (the Caballo timestep +3 days in the daily model) with an IF THEN ELSE statement, which checks if the BlwCourchesneLocalTimestep is after or equal to the date returned by the LRGDiversionStartDate function for EPCWID.  
If FALSE, the American Diversion Request gets set to 0 acre-ft.  
If TRUE, the American Diversion Request gets set to the value in the American Diversion row in the AnnualRHGDistribution tableslot multiplied by the end-of-month EPCWID Final Diversion Allocation multiplied by the value returned by an IF THEN ELSE statement which checks if the timestep is daily. If TRUE, the value returned is the value from the BlwCourchesneLocalTimestep row and the GetYearTypeasString column in the American DiversionPattern periodic tableslot. If FALSE, the value returned is the value returned by the Monthly Pattern function for the American DiversionPattern periodic tableslot (which aggregates the daily diversion pattern coefficients to monthly).  
The American Diversion Request is limited to the value in the Max Diversion slot on the American Diversion object.

If not already input, the sixth assignment sets the value on the Diversion Request slot on the AcequiaMadre diversion object at the BlwCourchesneLocalTimestep (the Caballo timestep +3 days in the daily model) to the value in the AcequiaMadre row in the AnnualRHGDistribution tableslot multiplied by the end-of-month Mexico Allocation multiplied by the value returned by an IF THEN ELSE statement which checks if the timestep is daily. If TRUE, the value returned is the value from the BlwCourchesneLocalTimestep row and the GetYearTypeasString column in the AcequiaMadreDiversionPattern periodic tableslot. If FALSE, the value returned is the value returned by the Monthly Pattern function for the AcequiaMadreDiversionPattern periodic tableslot (which aggregates the daily diversion pattern coefficients to monthly).  
The AcequiaMadre Diversion Request is limited to the value in the Max Diversion slot on the AcequiaMadre diversion object.  
This rule fires once per timestep if the Use Demand Based or Pattern Based Diversions switch on the Configuration object is set to 1 and if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function

Model slots written by rule:  
1. object.Diversion Request, where object is all 6 river headgate diversion objects in the LRG which divert from the river into the canals.

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Configuration  
3. Rio Grande Project Data  
4. American Diversion  
5. Eastside  
6. Westside  
7. LeasburgandPicacho  
8. AcequiaMadre  
9. CaballoToLeasburgTimeLag  
10. LeasburgToMesillaLag  
11. MesillaToAmericanLag  
12. D3 Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**EndOfMonth
* **RPL Object Icon**RinconLocalTimestep
* **RPL Object Icon**MesillaLocalTimestep
* **RPL Object Icon**GetYearTypeasString
* **RPL Object Icon**LeasburgLocalTimestep
* **RPL Object Icon**LRGDiversionStartDate
* **RPL Object Icon**Min
* **RPL Object Icon**GetTimestep
* **RPL Object Icon**VolumeToFlow
* **RPL Object Icon**IsInput
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**Monthly Pattern

### Control Display Icon2.2 RPL Object IconSet D3 Diversion Requests Based on Water User Demands

Rule Purpose:   
This rule sets the diversion requests for all Rio Grande Project river head gates based on actual on-farm diversion requests by each water user. These demands are scaled up to account for wasteway flows and canal seepage coefficient, and are limited to the corresponding district's allocation multiplied by a distribution coefficient.

Rule Logic: Execution Constraint logic is at end of explanation.  
A FOR DO statement is used to create a list of all the diversion objects in the All LRG River Head Gates subbasin, giving each diversion object the argument name Headgate. If not already input, the Diversion Request series slot for each Headgate is set at the LocalTimestep of the diversion object with an IF THEN ELSE statement, which checks if the current timestep is within the irrigation season, which is February through October, with reference to the user-defined IsIrrigationSeason function.   
If FALSE, the value for the Diversion Requested series slot on the Headgate is set equal to 0 acre-feet/month.

If TRUE, an internal IF THEN ELSE statement is used to check if the Headgate name is Eastside.   
If TRUE, the value for the Diversion Requested series slot on Eastside diversion object is set equal to the minimum of the Eastside diversion object Max Diversion or the value returned by the user-defined Get Eastside NM Diversion Requests plus the value returned by the Get Eastside TX Diversion Requests function, at the MesillaLocalTimestep.   
The Get Eastside NM Diversion Requests function returns the sum of the irrigation object Diversion Requested values in the Eastside NM region divided by 1 minus the Eastside Percent Waste divided by 1 minus the EastsideCanalSeepage fraction, limited to the value in the EastsideNM row in the AnnualRHGDistribution tableslot multiplied by EBID storage.  
The Get Eastside TX Diversion Requests function returns the sum of the irrigation object Diversion Requested values in the Eastside TX region divided by 1 minus the Eastside Percent Waste divided by 1 minus the EastsideCanalSeepage fraction, limited to the value in the EastsideTX row in the AnnualRHGDistribution tableslot multiplied by EPCWID storage.

If the Headgate name is not Eastside, then another internal IF THEN ELSE statement is used to check if the Headgate name is Westside.   
IF TRUE, the value for the Diversion Requested series slot on Westside diversion object is set in the same way that the Eastside Diversion Request was set.

If the Headgate name is not Westside, then another internal IF THEN ELSE statement is used to check if the Headgate name is ArreyandPercha.   
If TRUE, the value for the Diversion Requested series slot on ArreyandPercha diversion object is set equal to the value returned by the user-defined Get Rincon Diversion Requests function, at the RinconLocalTimestep.  
The Get Rincon Diversion Requests function returns the sum of the irrigation object Diversion Requested values in the Rincon region divided by 1 minus the Rincon Percent Waste divided by 1 minus the ArreyandPerchaSeepage fraction, limited to the value in the ArreyandPercha row in the AnnualRHGDistribution tableslot multiplied by EBID storage, and limited to the ArreyandPercha diversion object Max Diversion.

If the Headgate name is not ArreyandPercha, then another internal IF THEN ELSE statement is used to check if the Headgate name is LeasburgandPicacho.   
If TRUE, the value for the Diversion Requested series slot on LeasburgandPicacho diversion object is set in the same way that the ArreyandPercha Diversion Request was set.

If the Headgate name is not LeasburgandPicacho, then another internal IF THEN ELSE statement is used to check if the Headgate name is American Diversion.   
If TRUE, the value for the Diversion Requested series slot on the American Diversion object is set to the value returned by the user-defined GetBelowCourchesneDiversionRequests function, at the BlwCourchesneLocalTimestep.   
The GetBelowCourchesneDiversionRequests function returns the sum of the irrigation object Diversion Requested values below Courchesne Bridge divided by 1 minus the BelowCourchesne Percent Waste divided by 1 minus the EP1\_TotalCanalSeepage fraction, limited to the value in the American Diversion row in the AnnualRHGDistribution tableslot multiplied by EPCWID storage, and limited to the American Diversion object Max Diversion.

If the Headgate name is not American Diversion, then the Headgate name must be AcequiaMadre, and the value for the Diversion Requested series slot on the AcequiaMadre diversion object is set equal to the value returned by the user-defined Get Mexico Diversion Requests function, at the BlwCourchesneLocalTimestep.   
Within the Get Mexico Diversion Requests function, an IF THEN ELSE statement checks if any of the EBID or EPCWID river headgates have a non-zero Diversion Request.   
If FALSE, the Get Mexico Diversion Requests function returns 0 acre-feet.  
If TRUE, another internal IF THEN ELSE statement checks if the current timestep is equal to October.   
If TRUE, the Get Mexico Diversion Requests function returns the storage in the Mexico account on Caballo divided by the remaining days in October, limited to the AcequiaMadre diversion object Max Diversion.  
If FALSE, the Get Mexico Diversion Requests function returns the end-of-month Mexico Allocation multiplied by the value returned by an IF THEN ELSE statement which checks if the timestep is daily. If TRUE, the value returned is the value from the BlwCourchesneLocalTimestep row and the GetYearTypeasString column in the AcequiaMadreDiversionPattern periodic tableslot. If FALSE, the value returned is the value returned by the Monthly Pattern function for the AcequiaMadreDiversionPattern periodic tableslot (which aggregates the daily diversion pattern coefficients to monthly). This is limited to the AcequiaMadre diversion object Max Diversion.

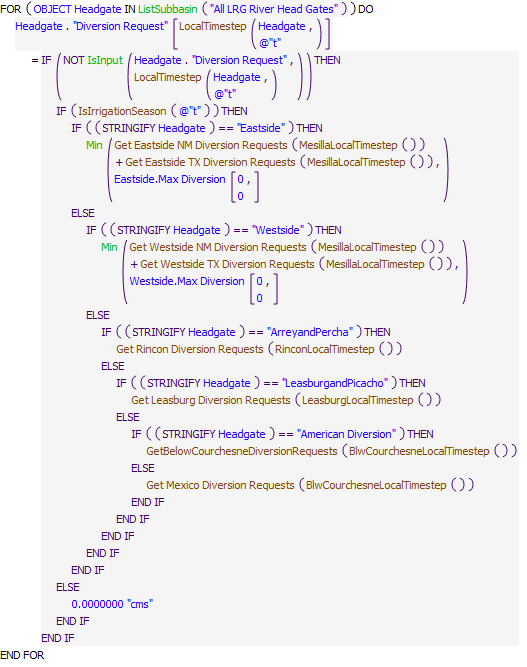
This rule fires once per timestep if the Use Demand Based or Pattern Based Diversions switch on the Configuration object is set to 0 and if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function

Model slots written by rule:  
1. object.Diversion Request, where object is all 6 river headgate diversion objects in the LRG which divert from the river into the canals.

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Configuration  
3. Rio Grande Project Data  
4. Caballo  
5. LRG Wasteway Data  
6. EP1\_TotalCanalSeepage  
7. American Diversion  
8. EastsideCanalSeepage  
9. Eastside  
10. WestsideCanalSeepage  
11. Westside  
12. LeasburgtoMesillaSeepage  
13. LeasburgandPicacho  
14. ArreyandPerchaSeepage  
15. AcequiaMadre  
16. CaballoToLeasburgTimeLag  
17. LeasburgToMesillaLag  
18. MesillaToAmericanLag  
20. D3 Data  
21. John Rogers WTP  
22. From Haskell WWTP  
23. Riverside Canal at Bustamante WWTP  
24. Robertson Umbenhauer WTP

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**GetBelowCourchesneDiversionRequests
* **RPL Object Icon**Get Leasburg Diversion Requests
* **RPL Object Icon**Get Mexico Diversion Requests
* **RPL Object Icon**Get Rincon Diversion Requests
* **RPL Object Icon**Get Eastside NM Diversion Requests
* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**RinconLocalTimestep
* **RPL Object Icon**MesillaLocalTimestep
* **RPL Object Icon**LocalTimestep
* **RPL Object Icon**LeasburgLocalTimestep
* **RPL Object Icon**Min
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**IsInput
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**Get Westside TX Diversion Requests
* **RPL Object Icon**Get Westside NM Diversion Requests
* **RPL Object Icon**IsIrrigationSeason
* **RPL Object Icon**Get Eastside TX Diversion Requests

### Control Display Icon2.3 RPL Object IconMexico Diversion 1 below Acequia Madre

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon2.4 RPL Object IconMexico Diversion 2 below Acequia Madre

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon2.5 RPL Object IconLimit EPCWID to Meet Mexico Demand

Rule Purpose:  
This rule sets the minimum bypass at American Diversion Dam to the equivalent of Mexico's demand. This guarantees that the Rio Grande Project delivery obligation to Mexico is always met.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the BlwCourchesneLocalTimestep value for the Minimum Diversion Bypass series slot on the RGatAmericanDiversionDam reach object equal to the value in the Diversion Request series slot on the AcequiaMadre diversion object at the BlwCourchesneLocalTimestep.   
This rule fires once per timestep.

Model slots written by rule:  
1. RGatAmericanDiversionDam.Minimum Diversion Bypass

List of key model objects with slots read by the rule or child functions:  
1. AcequiaMadre  
2. CaballoToLeasburgTimeLag  
3. LeasburgToMesillaLag  
4. MesillaToAmericanLag

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon2.6 RPL Object IconSet Rincon Wasteway Diversions

Rule Purpose:   
This rule sets the wasteway diversions in the Rincon basin using wasteway coefficient tables that allocate a percentage of the river headgate flow to wasteway flows, and coefficients that distribute the total wasteway flow across the individual wasteways in the Rincon basin.

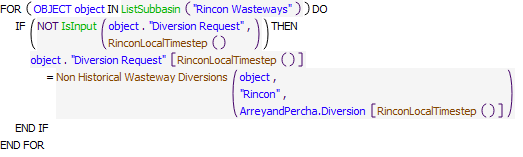
Rule Logic: Execution Constraint logic is at end of explanation.  
A FOR DO statement is used to create a list of all the diversion objects in the Rincon Wasteways subbasin. The RinconLocalTimestep values for the Diversion Requested series slots for each diversion object are set equal to the value returned by the user-defined Non Historical Wasteway Diversions function for the ArreyandPercha diversion object at the RinconLocalTimestep.   
The Non Historical Wasteway Diversions function returns the ArreyandPercha river headgate diversion, multiplied by the value in the Rincon Percent Waste scalar slot on the LRG Wasteway Data object, multiplied by the value in the RinconLocalTimestep row and the column corresponding to the specific wasteway diversion object in the Rincon Wasteway Splits tableslot on the LRG Wasteway Data object.  
This rule fires every timestep.

Model slots written by rule:  
1. Wasteway No 16.Diversion Request  
2. Wasteway No 18.Diversion Request  
3. Wasteway No 5.Diversion Request

List of key model objects with slots read by the rule or child functions:  
1. LRG Wasteway Data  
2. ArreyandPercha

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Referenced Functions

* **RPL Object Icon**RinconLocalTimestep
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**IsInput
* **RPL Object Icon**Non Historical Wasteway Diversions

### Control Display Icon2.7 RPL Object IconSet Rincon Area Diversion Requests

Rule Purpose:   
For each of the 5 subareas in the Rincon basin, this rule sets the diversion request at each subarea diversion equal to the irrigation demand in that subarea, limited by the total water diverted by the Arrey and Percha Diversion Dam river headgate with canal seepage and wasteway diversions subtracted, and split proportionally based on the area of that subarea relative to the total Rincon basin irrigated area. The irrigation demand is based on historical estimated consumptive irrigation requirement (CIR), farm efficiency, and historical irrigated acreage.

Rule Logic: Execution Constraint logic is at end of explanation.  
A FOR DO statement is used to create a list of all the subarea diversion objects in the Rincon Diversions subbasin. The RinconLocalTimestep value for the Diversion Requested series slot for each diversion object is set equal to the minimum of:   
1. The maximum of 0 acre-feet/month or the RinconLocalTimestep value in the Total Outflow series slot on the ArreyandPerchaSeepage canal object, minus the value returned by the user-defined function Sum Wasteway Diversions (which sums up the RinconLocalTimestep Diversion Request values for all the diversion objects in the Rincon Wasteways subbasin) multiplied by the value in the series slot corresponding to the subarea diversion object on the Rincon Percent Irrigated data object. This term limits the diversion request to a portion of the water available for diversion after canal seepage and wasteway flows have been subtracted.   
2. The RinconLocalTimestep value in the Diversion Requested series slots on the WaterUser object corresponding to the subarea diversion object. This value is computed based on crop CIR, crop area and on farm efficiency.  
This rule fires every timestep.

Model slots written by rule:  
1. object.Diversion Request, where object is all 5 diversion objects in the Rincon region which divert from the canal onto the farms.

List of key model objects with slots read by the rule or child functions:  
1. Rincon Percent Irrigated  
2. ArreyandPerchaSeepage

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Referenced Functions

* **RPL Object Icon**Sum Wasteway Diversions
* **RPL Object Icon**RinconLocalTimestep
* **RPL Object Icon**Min
* **RPL Object Icon**Max
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**GetObject
* **RPL Object Icon**Split

### Control Display Icon2.8 RPL Object IconSet Leasburg Wasteway Diversions

Rule Purpose:   
This rule sets the wasteway diversions in the Leasburg basin using wasteway coefficient tables that allocate a percentage of the river head gate flow to wasteway flows, and coefficients that distribute the total wasteway flow across the individual wasteways in the Leasburg basin.

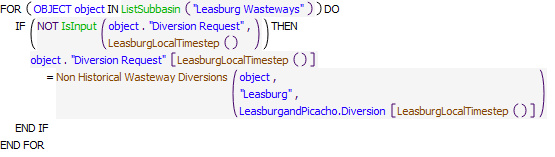
Rule Logic: Execution Constraint logic is at end of explanation.  
A FOR DO statement is used to create a list of all the diversion objects in the Leasburg Wasteways subbasin. The LeasburgLocalTimestep values for the Diversion Requested series slots for each diversion object are set equal to the value returned by the user-defined Non Historical Wasteway Diversions function for the LeasburgandPicacho diversion object at the LeasburgLocalTimestep.   
The Non Historical Wasteway Diversions function returns the LeasburgandPicacho river headgate diversion, multiplied by the value in the Leasburg Percent Waste scalar slot on the LRG Wasteway Data object, multiplied by the value in the LeasburgLocalTimestep row and the column corresponding to the specific wasteway diversion object in the Leasburg Wasteway Splits tableslot on the LRG Wasteway Data object.  
This rule fires every timestep.

Model slots written by rule:  
1. Wasteway No 1.Diversion Request  
2. Wasteway No 5 and 8.Diversion Request

List of key model objects with slots read by the rule or child functions:  
1. LRG Wasteway Data  
2. LeasburgandPicacho  
3. CaballoToLeasburgTimeLag  
4. LeasburgToMesillaLag

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Referenced Functions

* **RPL Object Icon**LeasburgLocalTimestep
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**IsInput
* **RPL Object Icon**Non Historical Wasteway Diversions

### Control Display Icon2.9 RPL Object IconSet Leasburg Area Diversion Requests

Rule Purpose:   
For each of the 5 subareas in the Leasburg basin, this rule sets the diversion request at each subarea headgate equal to the irrigation demand in that subarea, limited by the total water diverted by the Leasburg Diversion Dam river headgate with canal seepage and wasteway diversions subtracted, and split proportionally based on the area of that subarea relative to the total Leasburg basin irrigated area. The irrigation demand is based on historical estimated consumptive irrigation requirement (CIR), farm efficiency, and historical irrigated acreage.

Rule Logic: Execution Constraint logic is at end of explanation.  
A FOR DO statement is used to create a list of all the subarea diversion objects in the Leasburg Diversions subbasin. The LeasburgLocalTimestep value for the Diversion Requested series slot for each diversion object is set equal to the minimum of:   
1. The maximum of 0 acre-feet/month or the LeasburgLocalTimestep value in the Total Outflow series slot on the LeasburgtoMesillaSeepage canal object, minus the value returned by the user-defined function Sum Wasteway Diversions (which sums up the LeasburgLocalTimestep Diversion Request values for all the diversion objects in the Leasburg Wasteways subbasin) multiplied by the value in the series slot corresponding to the subarea diversion object on the Leasburg Percent Irrigated data object. This term limits the diversion request to a portion of the water available for diversion after canal seepage and wasteway flows have been subtracted.   
2. The LeasburgLocalTimestep value in the Diversion Requested series slots on the WaterUser object corresponding to the subarea diversion object. This value is computed based on crop CIR, crop area and on farm efficiency.   
This rule fires every timestep.

Model slots written by rule:  
1. object.Diversion Request, where object is all 5 diversion objects in the Leasburg region which divert from the canal onto the farms.

List of key model objects with slots read by the rule or child functions:  
1. Leasburg Percent Irrigated  
2. LeasburgtoMesillaSeepage  
3. CaballoToLeasburgTimeLag  
4. LeasburgToMesillaLag

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Referenced Functions

* **RPL Object Icon**Sum Wasteway Diversions
* **RPL Object Icon**LeasburgLocalTimestep
* **RPL Object Icon**Min
* **RPL Object Icon**Max
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**GetObject
* **RPL Object Icon**Split

### Control Display Icon2.10 RPL Object IconSet Mesilla Wasteway Diversions

Rule Purpose:   
This rule sets the wasteway diversions in the Mesilla basin using wasteway coefficient tables that allocate a percentage of the Eastside Mesilla or Westside Mesilla river head gate flows to wasteway flows, and coefficients that distribute the total wasteway flow across the individual wasteways on either the Eastside or Westside of the Mesilla basin.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule has two assignments. In the first assignment, a FOR DO statement is used to create a list of all the diversion objects in the Eastside Wasteways subbasin. The MesillaLocalTimestep values for the Diversion Requested series slots for each diversion object are set equal to the value returned by the user-defined Non Historical Wasteway Diversions function for Eastside Mesilla and the diversion object at the MesillaLocalTimestep.   
The Non Historical Wasteway Diversions function returns the Eastside river headgate diversion, multiplied by the value in the Eastside Percent Waste scalar slot on the LRG Wasteway Data object, multiplied by the value in the MesillaLocalTimestep row and the column corresponding to the specific wasteway diversion object in the Eastside Wasteway Splits tableslot on the LRG Wasteway Data object.  
In the second assignment, a FOR DO statement is used to create a list of all the diversion objects in the Westside Wasteways subbasin. The MesillaLocalTimestep values for the Diversion Requested series slots for each diversion object are set equal to the value returned by the user-defined Non Historical Wasteway Diversions function for Westside Mesilla and the diversion object at the MesillaLocalTimestep.   
This rule fires every timestep.

Model slots written by rule:  
1. object.Diversion Request where object is all 10 wasteway diversion objects in the Mesilla region which divert from the canal back to the river.

List of key model objects with slots read by the rule or child functions:  
1. LRG Wasteway Data  
2. Eastside  
3. Westside  
4. CaballoToLeasburgTimeLag  
5. LeasburgToMesillaLag

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Referenced Functions

* **RPL Object Icon**MesillaLocalTimestep
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**IsInput
* **RPL Object Icon**Non Historical Wasteway Diversions

### Control Display Icon2.11 RPL Object IconSet Mesilla to NM by Demand

Rule Purpose:   
This rule sets the diversions from Eastside and Westside "main" canals to New Mexico water users' canals based on the water available in the canal, less wasteway diversions, and multiplied by the proportion of New Mexico demands to the total demands on the canal.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule has two assignments. The first assignment sets the MesillaLocalTimestep value for the Mesilla Eastside to NM Diversion series slot on the Rio Grande Project Data object equal to the maximum of 0 acre-feet or the value returned by and IF THEN ELSE statement, which checks if the current timestep is within the irrigation season, which is February through October, with reference to the user-defined IsIrrigationSeason function, AND if the MesillaLocalTimestep value in the Total Outflow series slot on the Eastside Canal Loss object is larger than 0 acre-feet/month, AND if the value in the Storage series slot on the EBID account on the Caballo reservoir object on the previous timestep is larger than 0 acre-feet, AND if the sum of the values returned by the Get Eastside NM Diversion Requests and Get Eastside TX Diversion Requests functions at the Get Eastside NM Diversion Requests are more than zero (I.e., there are irrigation diversions in the Eastside area).   
IF TRUE, the Mesilla Eastside to NM Diversion is set equal to the MesillaLocalTimestep value in the Total Outflow series slot on the Eastside Canal Loss object minus the value returned by the user-defined Sum Wasteway Diversions function (which sums up the MesillaLocalTimestep Diversion Request values for all the diversion objects in the Eastside Wasteways subbasin), multiplied by the MesillaLocalTimestep value returned by the user-defined Get Eastside NM Diversion Requests divided by the value returned by the Get Eastside NM Diversion Requests function plus the value returned by the user-defined Get Eastside TX Diversion Requests function.

The second assignment sets the MesillaLocalTimestep value for the Mesilla Westside to NM Diversion series slot on the Rio Grande Project Data object equal to the maximum of 0 acre-feet or the value returned by and IF THEN ELSE statement, which checks if the current timestep is within the irrigation season, which is February through October, with reference to the user-defined IsIrrigationSeason function, AND if the MesillaLocalTimestep value in the Total Outflow series slot on the Westside Canal Loss object is larger than 0 acre-feet/month, AND if the value in the Storage series slot on the EBID account on the Caballo reservoir object on the previous timestep is larger than 0 acre-feet.   
IF TRUE, the Mesilla Westside to NM Diversion is set equal to the MesillaLocalTimestep value in the Total Outflow series slot on the Westside Canal Loss object minus the value returned by the user-defined Sum Wasteway Diversions function (which sums up the MesillaLocalTimestep Diversion Request values for all the diversion objects in the Westside Wasteways subbasin), multiplied by the MesillaLocalTimestep value returned by the user-defined Get Westside NM Diversion Requests function divided by the value returned by the Get Westside NM Diversion Requests function plus the value returned by the user-defined Get Westside TX Diversion Requests function.

This rule fires every timestep.

Model slots written by rule:  
1. Rio Grande Project Data.Mesilla Eastside to NM Diversion  
2. Rio Grande Project Data.Mesilla Westside to NM Diversion

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers   
2. Configuration  
3. LRG Wasteway Data  
4. CaballoToLeasburgTimeLag  
5. LeasburgToMesillaLag  
6. EastsideCanalSeepage  
7. WestsideCanalSeepage  
8. EP1\_TotalCanalSeepage  
9. Caballo

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Referenced Functions

* **RPL Object Icon**Get Eastside NM Diversion Requests
* **RPL Object Icon**Sum Wasteway Diversions
* **RPL Object Icon**IsHistoricalModelRun
* **RPL Object Icon**MesillaLocalTimestep
* **RPL Object Icon**Get Westside TX Diversion Requests
* **RPL Object Icon**Get Westside NM Diversion Requests
* **RPL Object Icon**IsIrrigationSeason
* **RPL Object Icon**Get Eastside TX Diversion Requests

### Control Display Icon2.12 RPL Object IconSet Mesilla Area Diversion Requests

Rule Purpose:   
For each subarea in the New Mexico portion of the Mesilla region, this rule sets the diversion request at each subarea diversion equal to the irrigation demand in that subarea, limited by the amount designated for New Mexico (as assigned in the previous rule), split proportionally based on the area of that subarea relative to the total Eastside or Westside NM irrigated area. The irrigation demand is based on historical estimated consumptive irrigation requirement (CIR), farm efficiency, and historical irrigated acreage.   
For each subarea in the Texas portion of the Mesilla region, this rule sets the diversion request at each subarea diversion equal to the irrigation demand in that subarea, limited by the total water diverted by the river headgate with canal seepage, wasteway diversions, and the portion designated for New Mexico subtracted, and split proportionally based on the area of that subarea relative to the total Eastside or Westside TX irrigated area. The irrigation demand is based on historical estimated consumptive irrigation requirement (CIR), farm efficiency, and historical irrigated acreage.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule contains four FOR DO statements.   
The first FOR DO statement is used to create a list of all the subarea diversion objects in the Eastside NM Diversions subbasin. The value for the Diversion Requested series slot for each diversion object is set equal to the minimum of:   
1. The MesillaLocalTimestep value in the Mesilla Eastside to NM Diversion series slot on the Rio Grande Project Data object, multiplied by the MesillaLocalTimestep value in the series slot corresponding to the subarea diversion object on the Eastside NM Percent Irrigated data object.   
2. The MesillaLocalTimestep value in the Diversion Requested series slots on the WaterUser object corresponding to the subarea diversion object. This value is computed based on crop CIR, crop area and on farm efficiency.

The second FOR DO statement is the same as the first, except it sets Diversion Requests for Westside NM Mesilla subarea diversion objects.

The third FOR DO statement is used to create a list of all the subarea diversion objects in the Eastside TX Diversions subbasin. The MesillaLocalTimestep value for the Diversion Requested series slot for each diversion object is set equal to the minimum of:   
1. The maximum of 0 acre-feet/month or the MesillaLocalTimestep value in the Total Outflow series slot on the Eastside Canal Loss canal object, minus the value returned by the user-defined function Sum Wasteway Diversions (which sums up the MesillaLocalTimestep Diversion Request values for all the diversion objects in the Eastside Wasteways subbasin), minus the MesillaLocalTimestep value in the Mesilla Eastside to NM Diversion series slot on the Rio Grande Project Data object, multiplied by the value in the scalar slot corresponding to the subarea diversion object on the Eastside TX Percent for Diversion data object.   
2. The MesillaLocalTimestep value in the Diversion Requested series slots on the WaterUser object corresponding to the subarea diversion object. This value is computed based on crop CIR, crop area and on farm efficiency.

The fourth FOR DO statement is the same as the third, except it sets Diversion Requests for Westside TX Mesilla subarea diversion objects.

This rule fires every timestep.

Model slots written by rule:  
1. object.Diversion Request, where object is all 11 diversion objects in the Mesilla region which divert from the canal onto the farms.

List of key model objects with slots read by the rule or child functions:  
1. Rio Grande Project Data  
2. Eastside NM Percent Irrigated  
3. Westside NM Percent Irrigated  
4. EastsideCanalSeepage  
5. WestsideCanalSeepage  
6. Eastside TX Percent for Diversion  
7. Westside TX Percent for Diversion  
8. CaballoToLeasburgTimeLag  
9. LeasburgToMesillaLag

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Referenced Functions

* **RPL Object Icon**Sum Wasteway Diversions
* **RPL Object Icon**MesillaLocalTimestep
* **RPL Object Icon**Min
* **RPL Object Icon**Max
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**GetObject
* **RPL Object Icon**Split

### Control Display Icon2.13 RPL Object IconMontoya Lateral Diversion Request

Rule Purpose:   
Just below the New Mexico-Texas state line, the Montoya Lateral takes water from West Texas Mesilla across the Rio Grande to East Texas Mesilla. The Montoya lateral serves the subareas S20 and S21, as well as wasteways 36 and 38.

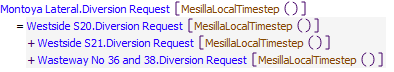
Rule Logic:  
This rule sets the MesillaLocalTimestep value for the Diversion Request series slot on the Montoya Lateral object equal to the sum of the MesillaLocalTimestep values on the Diversion Request series slots on the Westside 20, Westside 21 and Wasteway No 36 and 38 diversion objects.   
This rule fires every timestep.

Model slots written by rule:  
1. Montoya Lateral.Diversion Request

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Westside S20  
3. Westside S21  
4. Wasteway No 36 and 38  
5. CaballoToLeasburgTimeLag  
6. LeasburgToMesillaLag

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Referenced Functions

* **RPL Object Icon**MesillaLocalTimestep

### Control Display Icon2.14 RPL Object IconSplit Ag Pumping between Deep and Shallow Aquifers

Rule Purpose:   
This rule sets the portion of groundwater pumping coming from the Shallow Aquifer (as opposed to the Deep Aquifer) for every supplemental groundwater user and primary groundwater user in the Mesilla and Leasburg areas, besides subarea S21 (this is not necessary because there is no Deep Aquifer modeled here). The split between Shallow and Deep Aquifer pumping is modeled using a Pipe Junction object, with the shallow aquifer pumping connected to the first inflow, and the deep aquifer pumping connected to the second inflow. The first inflow is connected to the underlying alluvial aquifer object, while the second inflow is not connected to anything but represents inflow from a deep aquifer that is not explicitly modeled.

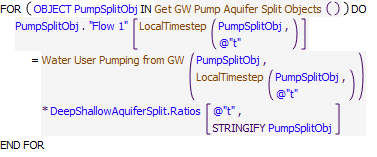
Rule Logic: Execution Constraint logic is at end of explanation.  
For every pipe junction object in URGWOM that has the phrase "AquiferSplit" in it, this rule sets the value for the Flow1 series slot on the pipe junction object equal to the value in the Diversion series slot on the corresponding primary groundwater water user object, or the value in the Supplemental Diversion series slot on the corresponding supplemental groundwater water user object, multiplied by the value from the Ratios table slot on the DeepShallowAquiferSplit object that corresponds to the water user object. This rule fires on every timestep, and sets values at the Local Timestep (the Caballo timestep + the lag time down to the water user object).  
This rule fires every timestep.

Model slots written by rule:  
1. PumpSplitObj.Flow1 where PumpSplitOpj is all 28 PipeJunction objects in the Leasburg and Mesilla regions which split groundwater pumping between the deep and shallow aquifer layers, and Flow1 is the flow coming from the shallow aquifer layer.

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. DeepShallowAquiferSplit

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Referenced Functions

* **RPL Object Icon**Get GW Pump Aquifer Split Objects
* **RPL Object Icon**LocalTimestep
* **RPL Object Icon**Water User Pumping from GW

### Control Display Icon2.15 RPL Object IconSet BelowCourchesne Wasteway Diversions

Rule Purpose:   
This rule sets the wasteway diversions in the Below Courchesne region using wasteway coefficient tables that allocate a percentage of the river headgate flow to wasteway flows, and coefficients that distribute the total wasteway flow across the individual wasteways in that basin.

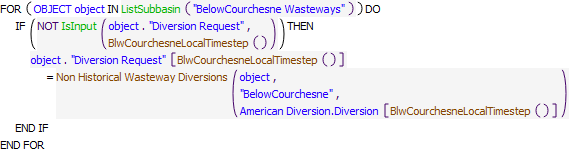
Rule Logic: Execution Constraint logic is at end of explanation.  
A FOR DO statement is used to create a list of all the diversion objects in the American Wasteways subbasin. The BlwCourchesneLocalTimestep values for the Diversion Requested series slots for each diversion object are set equal to the value returned by the user-defined Non Historical Wasteway Diversions function for the Below Courchesne region and the diversion object at the BlwCourchesneLocalTimestep.  
The Non Historical Wasteway Diversions function returns the American river headgate diversion, multiplied by the value in the BelowCourchesne Percent Waste scalar slot on the LRG Wasteway Data object, multiplied by the value in the BlwCourchesneLocalTimestep row and the column corresponding to the specific wasteway diversion object in the BelowCourchesne Wasteway Splits tableslot on the LRG Wasteway Data object.  
This rule fires every timestep.

Model slots written by rule:  
1. Ascarate Wasteway.Diversion Request  
2. Fabens Waste Channel to RG.Diversion Request  
3. Riverside Wasteway No 1.Diversion Request  
4. Riverside Wasteway No 2.Diversion Request

List of key model objects with slots read by the rule or child functions:  
1. American Diversion  
2. CaballoToLeasburgTimeLag  
3. LeasburgToMesillaLag  
4. MesillaToAmericanLag

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

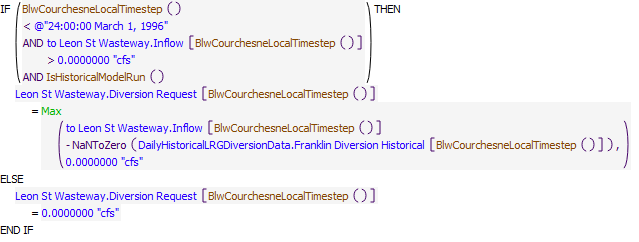
****

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**IsInput
* **RPL Object Icon**Non Historical Wasteway Diversions

### Control Display Icon2.16 RPL Object IconSet Leon St Wasteway

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**IsHistoricalModelRun
* **RPL Object Icon**Max
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon2.17 RPL Object IconSet BelowCourchesne Area Diversion Requests

Rule Purpose:   
For each of the subareas in the American region this rule sets the diversion request at each subarea headgate equal to the irrigation demand in that subarea increased for the canal seepage in that subarea, limited by the total water diverted by the American Diversion Dam river headgate with wasteway diversions subtracted, water treatment plant diversions subtracted, and wastewater treatment plant returns added, and split proportionally based on the irrigation demand in that subarea relative to the total irrigation demand in the American region.

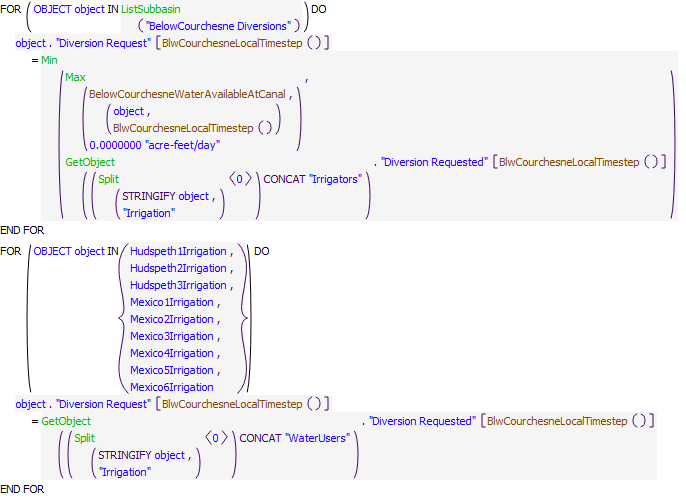
Rule Logic: Execution Constraint logic is at end of explanation.  
A FOR DO statement is used to create a list of all the subarea diversion objects in the American Diversions subbasin. The value for the Diversion Requested series slot for each diversion object is set equal to the minimum of:   
1. The value returned by the user defined function WaterAvailableFromAmericanDiversion for the subarea diversion object.   
2. The value in the Diversion Requested series slots on the water user object connected to the subarea diversion object.  
The rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. FranklinIrrigation.Diversion Request  
2. Franklin2Irrigation.Diversion Request  
3. RiversideIrrigation.Diversion Request  
4. TornilloIrrigation.Diversion Request

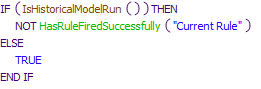
List of key model objects with slots read by the rule or child functions:  
1. CaballoToLeasburgTimeLag  
2. LeasburgToMesillaLag  
3. MesillaToAmericanLag  
4. EP1\_TotalCanalSeepage  
5. From Haskell WWTP  
6. Riverside Canal at Bustamante WWTP  
7. John Rogers WTP  
8. BelowCourchesne Percent Irrigated

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.  
8/23/2018: Marc Sidlow. Added Execution Constraint "NOT HasRuleFiredSuccessfully ("Current Rule")"  
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

****

Referenced Functions

* **RPL Object Icon**BelowCourchesneWaterAvailableAtCanal
* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**IsHistoricalModelRun
* **RPL Object Icon**Min
* **RPL Object Icon**Max
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**GetObject
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**Split

### Control Display Icon2.18 RPL Object IconSet Franklin Diversion Requests

Rule Purpose:   
This rule sets the demand for the Franklin canal headgate equal to the sum of the FranklinIrrigation and Ascarate wasteway demands, and the Franklin Feeder Diversion to the irrigation demand at Franklin2Irrigation. The purpose of these assignments are to move water into the Franklin canal as it approximately would have been moved historically. This rule also sets the Riverside Coffer Dam diversion to zero, since Riverside diversion no longer exists. This rule also sets the pumping and spill between the Fabens Waste Drain and Tornillo Canal to zero, and from Tornillo Canal into Fabens Waste Channel to zero, since very limited historical data exists.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule has six assignments. The first assignment sets the BlwCourchesneLocalTimestep value for the Diversion Request series slot on the Franklin Diversion object equal to the sum of the BlwCourchesneLocalTimestep values in the Diversion Request series slots on the FranklinIrrigation and Ascarate Wasteway diversion objects.   
The second assignment sets the BlwCourchesneLocalTimestep value for the Diversion Request series slot on the Franklin Feeder Diversion object equal to the BlwCourchesneLocalTimestep value in the Diversion Request series slots on the Franklin2Irrigation object.  
The third through sixth assignments set the values for the Diversion Request series slots on the Riverside Coffer Dam, Fabens Waste Drain Pump to Tornillo Canal, Tornillo Canal Spill to Fabens Waste Drain, and Tornillo Canal Spill to Fabens Waste Channel diversion objects equal to zero acre-feet.  
This rule fires every timestep.

Model slots written by rule:  
1. Franklin Diversion.Diversion Request  
2. Franklin Feeder Diversion.Diversion Request  
3. Riverside Coffer Dam.Diversion Request  
4. Fabens Waste Drain Pump to Tornillo Canal.Diversion Request  
5. Tornillo Canal Spill to Fabens Waste Drain.Diversion Request  
6. Tornillo Canal Spill to Fabens Waste Channel.Diversion Request

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. FranklinIrrigation  
3. Franklin2Irrigation  
4. Ascarate Wasteway  
5. CaballoToLeasburgTimeLag  
6. LeasburgToMesillaLag  
7. MesillaToAmericanLag

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**IsHistoricalModelRun
* **RPL Object Icon**Max

### Control Display Icon2.19 RPL Object IconSet Fabens Waste Drain Pumping and Spill

The pumping from the Fabens Waste Drain into the Tornillo Canal (or the spill visa-versa) is set equal to the difference in flow between Fabens Waste Drain and the historical Fabens Waste Channel if data exists for the latter, and zero otherwise

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**IsHistoricalModelRun
* **RPL Object Icon**Min
* **RPL Object Icon**Max
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon2.20 RPL Object IconSet Tornillo Canal Spill to Fabens Waste Channel

The pumping from the Fabens Waste Drain into the Tornillo Canal (or the spill visa-versa) is set equal to the difference in flow between Fabens Waste Drain and the historical Fabens Waste Channel if data exists for the latter, and zero otherwise

Statements

****

Execution Constraint

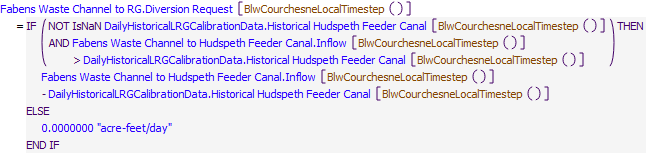
**Execution Constraint**

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**IsHistoricalModelRun
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon2.21 RPL Object IconSet Fabens Waste Channel to RG

Statements

****

Execution Constraint

**Execution Constraint**

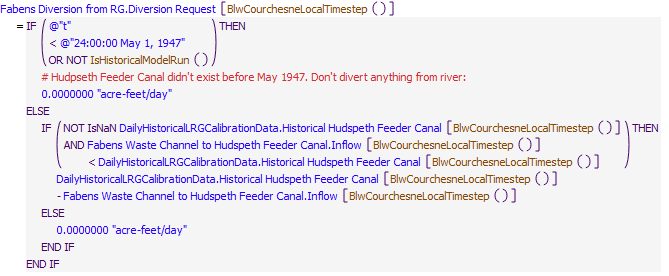
Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**IsHistoricalModelRun
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon2.22 RPL Object IconSet Fabens Diversion from RG

The pumping from the Fabens Waste Channel into the Rio Grande is set equal to the difference in flow between the (Fabens Waste Channel plus the Tornillo Canal Spill into the Fabens Waste Channel) and the historical Hudspeth Feeder Canal, if data for the latter exists.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**IsHistoricalModelRun
* **RPL Object Icon**HasRuleFiredSuccessfully

## Control Display Icon3 RPL Object IconLRG Reservoir Operations

### Control Display Icon3.1 RPL Object IconSet EB Pattern Based Release

Rule Purpose:   
This rule sets the release from Elephant Butte reservoir such that the pool elevation of Caballo reservoir does not fall below its target storage level given downstream demands, and initial storage conditions. This rule will increase the release Elephant Butte reservoir if there is additional generator release capacity, based on the number of generators online, and the Elephant Butte operating head.  
If Caballo reservoir has sufficient storage to meet demands and not fall below its minimum storage, Elephant Butte reservoir will release no water.

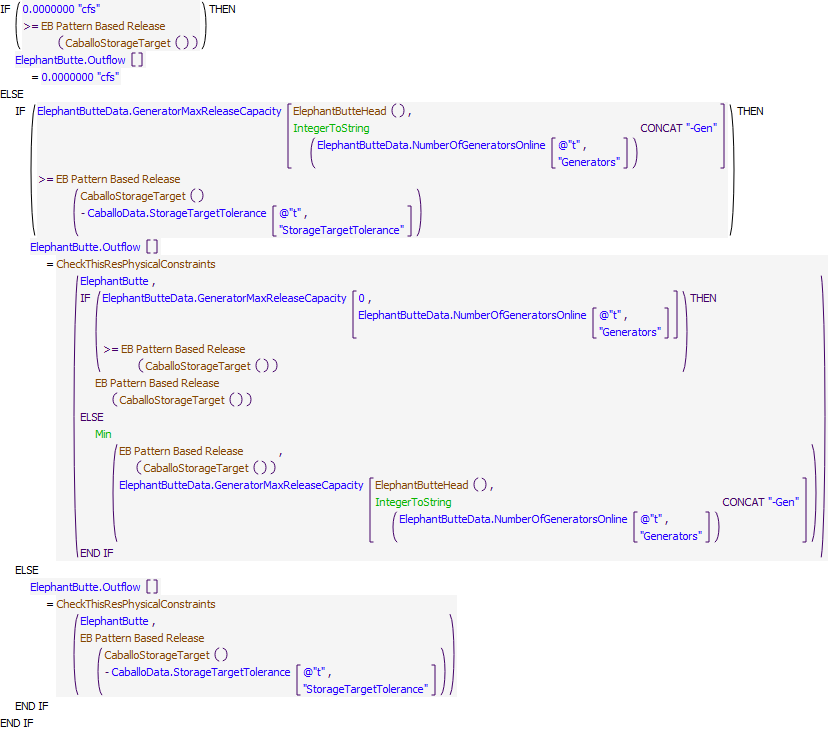
Rule Logic: Execution Constraint logic is at end of explanation.  
IF the value returned by the user-defined EB Pattern Based Release function, given the CaballoStorageTarget, is less than or equal to 0, this rule sets the value for the Outflow series slot on the ElephantButte reservoir object equal to 0 cfs.  
ELSE, this rule sets the value for the Outflow series slot on the ElephantButte reservoir with an IF THEN ELSE statement which checks if the Elephant Butte GeneratorReleaseCapacity, given the ElephantButteHead level (a level based on the Operating Head) and the NumberOfGeneratorsOnline, is larger than the value returned by the EB Pattern Based Release function, given the CaballoStorageTarget minus a StorageTargetTolerance.  
If FALSE, the ElephantButte Outflow is set equal to the value returned by the EB Pattern Based Release function, given the CaballoStorageTarget minus a StorageTargetTolerance. The outflow is constrained by the CheckThisResPhysicalConstraints function to be no less that the minimum required spill (there is no minimum required spill for Elephant Butte because its spillway is Regulated), to be no more than the maximum possible outflow (given the inflow, the turbine release capacity, and the spillway capacity), and to not drawdown Elephant Butte below the ElephantButteData.Minimum Storage.  
If TRUE, an internal IF THEN ELSE statement checks if the Elephant Butte GeneratorReleaseCapacity, given a ElephantButteHead level of 0 and the NumberOfGeneratorsOnline, is larger than the value returned by the EB Pattern Based Release function, given the CaballoStorageTarget.  
If TRUE, the ElephantButte Outflow is set equal to the value returned by the EB Pattern Based Release function, given the CaballoStorageTarget. The outflow is constrained by the CheckThisResPhysicalConstraints function.  
The EB Pattern Based Release function returns the minimum of the maximum possible Elephant Butte outflow (given the inflow, the turbine release capacity, and the spillway capacity) or the computed inflow to Caballo, given the Caballo release (as computed by the Caballo Pattern Based Release function), the CaballoStorageTarget, and the previous Caballo storage.   
If FALSE, the ElephantButte Outflow is set equal to the miniumum of the value returned by the EB Pattern Based Release function, given the CaballoStorageTarget, or the Elephant Butte GeneratorReleaseCapacity, given the ElephantButteHead level and the NumberOfGeneratorsOnline. The outflow is constrained by the CheckThisResPhysicalConstraints function.  
The rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function, and if the Use Demand Based or Pattern Based Diversions switch on the Configuration object is set to 1.

Model slots written by rule:  
1. ElephantButte.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. ElephantButte  
3. ElephantButteData  
4. Caballo  
5. CaballoData  
6. Configuration

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**CheckThisResPhysicalConstraints
* **RPL Object Icon**CaballoStorageTarget
* **RPL Object Icon**ElephantButteHead
* **RPL Object Icon**EB Pattern Based Release
* **RPL Object Icon**Min
* **RPL Object Icon**IntegerToString
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon3.2 RPL Object IconSet EB Release

Rule Purpose:   
This rule sets the release from Elephant Butte reservoir such that the pool elevation of Caballo reservoir does not fall below its target storage level given downstream demands, and initial storage conditions. The release from Elephant Butte is limited to the Rio Grande project storage in the reservoir, the physical storage in the reservoir minus a minimum storage, and the maximum possible outflow (given the inflow, the turbine release capacity, and the spillway capacity).

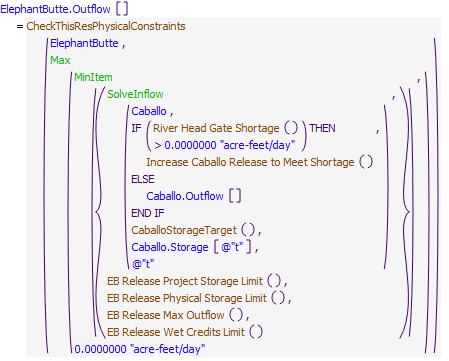
Rule Logic:  
This rule sets the value for the Outflow series slot on the ElephantButte reservoir object equal to the maximum of 0 acre-feet/month or the minimum of:   
1. The value returned by the predefined SolveInflow function given the reservoir object for which to calculate is the Caballo reservoir object, the end of timestep storage is the value in the StorageTarget periodic tableslot on the CaballoData object for the current timestep, the beginning of timestep storage is the value in the Storage series slot on the Caballo reservoir object on the previous timestep, and the average outflow over the timestep is the value returned by an IF THEN ELSE statement, which checks if the value returned by the user-defined River Head Gate Shortage function is larger than 0 acre-feet/month. If TRUE, then the average outflow over the timestep is the value returned by the user-defined Increase Caballo Release to Meet Shortage function. If FALSE, the average outflow over the timestep is the value in the Outflow series slot on the Caballo reservoir object.  
The Increase Caballo Release to Meet Shortage function returns the Caballo Outflow plus 75% of the sum of the shortages at all of the LRG river headgates, plus a buffer of 10 acre-feet.  
2. The value returned by the EB Release Project Storage Limit function. This function returns the Rio Grande Storage on Elephant Butte as a flow plus the Elephant Butte Inflow.  
3. The value returned by the EB Release Physical Storage Limit function. This function returns the physical Elephant Butte storage minus the ElephantButteData.Minimum Storage.  
4. The value returned by the EB Release Max Outflow function. This function returns the maximum possible Elephant Butte outflow (given the inflow, the turbine release capacity, and the spillway capacity).  
5. The value returned by the EB Release Wet Credits Limit function. This function returns the max outflow such that the Elephant Butte physical storage doesn't drop below the sum of the non-Rio Grande Project account storages on Elephant Butte.   
The rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function, if the current timestep is after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay, and if the Use Demand Based or Pattern Based Diversions switch on the Configuration object is set to 0.

Model slots written by rule:  
1. ElephantButte.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. ElephantButte  
3. ElephantButteData  
4. Caballo  
5. CaballoData  
6. Configuration

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**CheckThisResPhysicalConstraints
* **RPL Object Icon**CaballoStorageTarget
* **RPL Object Icon**EB Release Max Outflow
* **RPL Object Icon**EB Release Wet Credits Limit
* **RPL Object Icon**EB Release Project Storage Limit
* **RPL Object Icon**EB Release Physical Storage Limit
* **RPL Object Icon**MinItem
* **RPL Object Icon**Max
* **RPL Object Icon**SolveInflow
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate
* **RPL Object Icon**River Head Gate Shortage
* **RPL Object Icon**Increase Caballo Release to Meet Shortage

### Control Display Icon3.3 RPL Object IconZero EB Release For Power Optimization

Rule Purpose:   
This rule sets ElephantButte.Outflow to 0 cfs if the ElephantButteData.EBReleaseSeason = 0 (not releasing). This is to override any values set from previous rules during these days.

Rule Logic:  
If Configuration.EBPowerOptimization equals 1 and ElephantButteData.EBReleaseSchedule equals 0, then ElephantButte.Outflow equals 0 cfs.

Model slots written by rule:  
1. ElephantButte.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ElephantButteData

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
7/30/20: Lucas Barrett. Created Rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon3.4 RPL Object IconElephant Butte Release For Power Optimization

Rule Purpose:   
This rule only fires when EBPowerOptimization is turned on. When the rule, Elephant Butte Release For Power Optimization, fires, the rule will either calculate the flow needed for a new optimum power for Elephant Butte or calculate the flow needed to continue to generate the previous days optimum power. The rule will calculate a new optimum power when the following three criteria are met: ElephantButteData.EBReleaseSeason = 1, is not Saturday or Sunday, and the MW generated has not changed in the last specified number days. The reason for having no changes on the weekends is that generally, Reclamation does not make reservoir changes at Elephant Butte on the weekends unless for an emergency. The reason for having a specified number of days before another change in power generated is that Elephant Butte Field Office generally does not want to continually make changes to Elephant Butte releases. There is also the coordination with WAPA where WAPA needs at least a two to three full days? notice before Elephant Butted changes the MW generated. This time period is set to two weeks, but can be changed based on user discretion in the ElephantButteData.DaysUntilEBCanChangeMW. To be realistic with WAPA requirements, the number of days this is set to shouldn?t go below 3 days. When the rule calculates a new optimum generation, the rule will look out the next specified days? worth of flows in the CaballoData.TotalOutflowDirectlyInputToOverrideRules and find a daily generation that will produce a flow that will keep Caballo around the targeted storage, while taking into account the maximum and minimum flows that can be released out of the generators for the given operational head. The rule will use a 3D interpolation of that flow using the ElephantButte.PowerPlantTable and current Elephant Butte Operational Head to come up with a corresponding MW, then round that MW up to the nearest whole MW, and then do another 3D interpolation to calculate the corresponding flow to that MW. An additional if statement was added so that if the change in flows between the previously calculated flow and the new flow is more than half the previous flow or if the new specified day flow is less than 0 cfs, the new flow will instead be the previous 10 day flow divided by 2. This calculation is to remove any dramatic decreases in Elephant Butte outflows due to a few extremely low days or a change in the Caballo target due to Elephant Butte storage hitting one of the limits. One important note is that this rule will also always calculate to the nearest full MW, rounded up, as WAPA does not take anything less than a full megawatt. If in irrigation season, but either the current timestep is on a weekend or the specified amount of days being the same generation has not been met, the rule will calculate a new flow using a 3D interpolation of the previous day?s head and the MW that was calculate for the period.

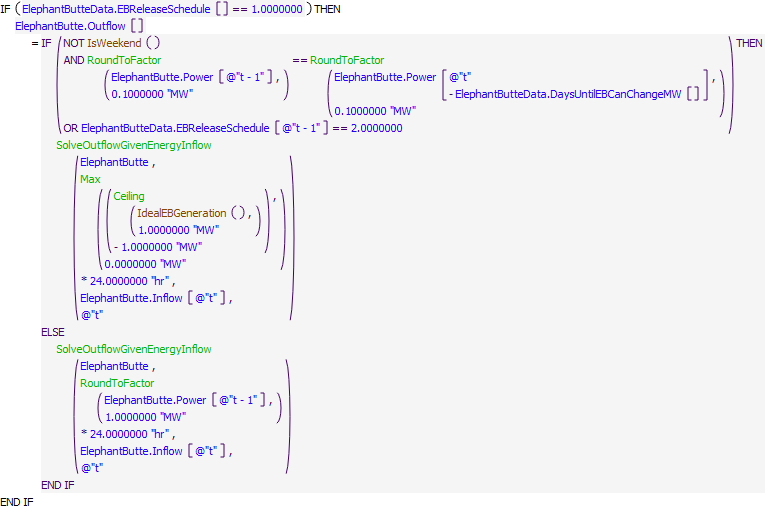
Rule Logic:  
If Configuration.EBPowerOptimization equals 1, and ElephantButteData.EBReleaseSchedule equals 1 then if not weekend and the previous days ElephantButte.Power equals ElephantButte.Power back a specified number of days (ElephantButteData.DaysUntilEBCanChangeMW) or the previous day's ElephantButteData.EBReleaseSchedule equals 2, ElephantButte.Outflow equals the flow required to meet the ceiling of the IdealEBGeneration to the nearest MW else ElephantButte.Outflow equals the flow required to meet the previous day's rounded ElephantButte.Power.

Model slots written by rule:  
1. ElephantButte.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ElephantButteData  
2. ElephantButte  
3. CaballoData  
4. Caballo

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
7/30/20: Lucas Barrett. Created Rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**IdealEBGeneration
* **RPL Object Icon**IsWeekend
* **RPL Object Icon**Ceiling
* **RPL Object Icon**Max
* **RPL Object Icon**RoundToFactor
* **RPL Object Icon**SolveOutflowGivenEnergyInflow
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon3.5 RPL Object IconPreload and Lower Caballlo

Rule Purpose:   
This rule only fires when EBPowerOptimization is turned on. This rule has two tasks. First it will preload Caballo starting on a specified amount of days (ElephantButteData.DaysOfReleaseToPreLoadCaballo) before irrigation season starts. If that date lands on a weekend, then it will wait until the first day that isn?t a weekend. For the first day, the rule will analyze how much Elephant Butte needs to release to get Caballo up to the determined CaballoData.StorageTargetsForEBPowerOptimization target, and using the current operating head, find the nearest whole MW, rounded up, that will produce a flow closest to the flow needed to get Caballo to that target. The following days up until the start of Caballo releases, this rule will calculate the flow needed to stay at the MW calculated on the first day. As mentioned previously, this preload period is set to a default of two weeks, based on current operations, but can be modified.   
The second part of this rule is to start drawing down Caballo at the end of irrigation season. The rule will begin firing at a specified (ElephantButteData.DaysToDrawDownCaballo) period before the end of irrigation season. The rule will calculate what is required to draw down Caballo so that Caballo will be close to the winter target by the last day of Caballo releases. For every specified number of days (ElephantButteData.DaysUntilEBCanChangeMW), this rule will recalculate using the previous day?s storage minus the winter storage target plus the sum of flows for the specified period to come up with an average flow needed over that specified period. The rule will then use a 3D interpolation of that flow using the ElephantButte.PowerPlantTable and current Elephant Butte Operational Head to come up with a corresponding MW, then round that MW up to the nearest whole MW, and then do another 3D interpolation to calculate the corresponding flow to that MW. For everyday between the DaysUntilEBCanChangeMW, the rule will calculate a new flow using a 3D interpolation of the previous day?s head and the MW that was calculate for this period. This rule will also tell Elephant Butte outflow to go to 0 starting on the last day of irrigation season, as generally, Elephant Butte shuts off one day before Caballo because of the time lag for water to get to Caballo.   
Rule Logic:  
If Configuration.EBPowerOptimization equals 1 and ElephantButteData.EBReleaseSchedule equals 2 then if Elephant Butte.Outflow from the previous timestep is greater than 5 cfs or it is the weekend then ElephantButte.Outflow equals the flow needed to keep the same generation to the whole MW as the previous day. If ElephantButte.Outlow is less than 5 cfs and not the weekened then ElephantButte.Outflow equals the flow from the calculated generation by a 3D interpolation of the ElephantButte.Plant Power Table that has a calculated average flow between the start of Elephant Butte releases and the start of Caballo releases (CaballoData.BeginCaballoReleaseDate) needed to reach Caballo's irrigation target storage that is between the MaxGeneratorFlowForHead and MinGeneratorFlowForHead. If the flow calculated is less than the MinGeneratorFlowForHead the gernerators are set to 0 MW which results in 0 cfs outflows, while if the flow is greater than the MaxGeneratorFlowForHead, the flow will be set to the MaxGeneratorFlowForHead.  
If Configuration.EBPowerOptimization equals 1 and ElephantButteData.EBReleaseSchedule equals 3 then if not the weekend and the generatation between the previous timestep and the timestep back a specified number of days (ElephantButteData.DaysUntilEBCanChangeMW) are equal and the current timestep does not equal CaballoData.EndCaballoReleaseData and if the calculated flow needed for Caballo storage to reach Caballo's winter season target based on the sum of CaballoData.TotalOutflowDirectlyInputToOverrideRules and the current Caballo storage is greater than 0 cfs then ElephantButte.Outflow equals EBIdealOutflowForCaballoDrawdown. If the calculated flow needed for Caballo storage to reach Caballo's winter season target is less than 0 cfs, that ElephantButte.Outflow equals 0 cfs. If is weekend or the generatation between the previous timestep and the timestep back a specified number of days (ElephantButteData.DaysUntilEBCanChangeMW) are not equal and if the current timestep is equal to CaballoData.EndCaballoReleaseDate then ElephantButte.Outflow is 0 cfs. Otherwise, if the current timestep is not equal to CaballoData.EndCaballoReleaseDate and EBIDealOutflowForCaballoDrawdown is greater than the previous ElehpantButte.Outflow \* 2, ElephantButte.Outflow equals EBIdealOutflowForCaballoDrawdown else if it is less than the previous ElephantButte.Outflow \* 2 then ElephantButte.Outflow equals the flow required to keep the previous ElephantButte.Power the same.

Model slots written by rule:  
1. ElephantButte.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ElephantButteData  
2. ElephantButte  
3. CaballoData  
4. Caballo

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
7/30/20: Lucas Barrett. Created Rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**MaxGeneratorFlowForHead
* **RPL Object Icon**CaballoStorageTargetEBPowerOptimization
* **RPL Object Icon**MinGeneratorFlowForHead
* **RPL Object Icon**EBIdealOutflowForCaballoDrawdown
* **RPL Object Icon**IsWeekend
* **RPL Object Icon**Min
* **RPL Object Icon**Ceiling
* **RPL Object Icon**SumSlot
* **RPL Object Icon**RoundToFactor
* **RPL Object Icon**NumberToDate
* **RPL Object Icon**GetDayOfYear
* **RPL Object Icon**SolveOutflowGivenEnergyInflow
* **RPL Object Icon**VolumeToFlow
* **RPL Object Icon**TableInterpolation3D
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon3.6 RPL Object IconEB Release In Between Generators

Rule Purpose:   
This rule only fires when EBPowerOptimization is turned on. Because calculating the ideal power generation and the flow that is required for that generation requires interpolation, there are times that the interpolation calculates a MW that is between zero to one, one to two, or two to three generators. In this case, the model starts producing incorrect numbers for flow to meet this power generation. To eliminate this issue, this rule looks to see if the previous rules calculated a MW that will have this issue and rounds down to the nearest MW, instead of rounding up which is done in the IdealEBGenerattion function, or subtracts 1 MW and then interprets the new flow to remedy this issue.

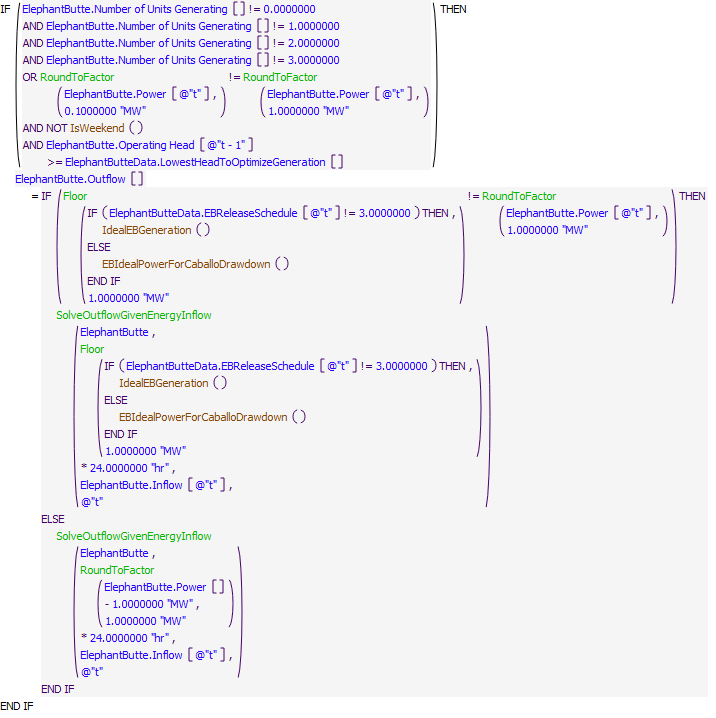
Rule Logic:   
If Configuration.EBPowerOptimization = 1, it is not the weekend, Elephant Butte.OperatingHead for the previous timestep is greater than or equal to the ElephantButteData.LowestOperatingHead (default = 90ft), and either the ElephantButte.Number of Units Generating does not equal a whole number or ElephantButte.Power rounded to 0.1 MW is not equal to a whole number then ElephantButte.Outflow equals the flow equivalent to the IdealEBGeneration rounded down to the nearest MW if the IdealEBGeneration rounded down to the nearest MW does not equal ElephantButte.Power calculated from the previous rules rounded to the nearest whole MW. Otherwise, ElephantButte.Outflow equals the equivalent flow required for ElephantButte.Power calculated from the previous rules minus 1 MW rounded to the nearest MW.

Model slots written by rule:  
1. ElephantButte.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ElephantButteData  
2. ElephantButte  
3. CaballoData  
4. Caballo

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
7/30/20: Lucas Barrett. Created Rule.  
2/28/23: Added if/else statement to use IdealEBGeneration or EBIdealPowerForCaballoDrawdown function depending of if it is during the drawdown period or not.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**IdealEBGeneration
* **RPL Object Icon**EBIdealPowerForCaballoDrawdown
* **RPL Object Icon**IsWeekend
* **RPL Object Icon**Floor
* **RPL Object Icon**RoundToFactor
* **RPL Object Icon**SolveOutflowGivenEnergyInflow
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon3.7 RPL Object IconEB Release To Stay Off Archeological Sites

This rule only fires when EBPowerOptimization is turned on. Reclamation does its best to avoid having the Caballo?s elevation fluctuate up and down over the two archeological sites in Caballo reservoir (Puskas and Animas). Doing so increases the chance of damage and erosion to the site. This rule was created with the purpose of ensuring that once Caballo elevation went below an the Animas site, it would stay below that site for the remainder of the season until the preloading of the next season and it would attempt to stay above the Puskas site. This resulted in a rule with two separate if statements (one for each site) that ensured once below the Animas site, Elephant Butte would use a lower target for Caballo as to not go back above the site while still maintaining above the Puskas site.   
Rule Logic:  
The rule first looks if the Caballo.Pool Elevation in the previous timestep is less than the elevation plus buffer of the Puskas site in the CaballoData.CaballoSites slot and if the EBReleaseSchedule is equal to 1 or 3 (release to meet target or drawdown) and if the new outflow that would be calculated is greater than the current calculated outflow (i.e., if the outflow calculated by energy input from the ceiling of the MWs interpolated from the ElephantButte.Plant Power Table using the minimum of Caballo release [CaballoData.TotalOutflowDirectlyInputToOverrideRules] minus ElephantButteToCaballo.Total GainLoss of the previous timestep minus ElephantButteToCaballoLocalInflow.Local Inflow and the maximum flow at of the generator per current head is greater than the current calculated outflow). If all that is true a new ElephantButte.Outflow will be calculated by the energy input from the ceiling of the MWs interpolated from the ElephantButte.Plant Power Table using the minimum of Caballo release [CaballoData.TotalOutflowDirectlyInputToOverrideRules] minus ElephantButteToCaballo.Total GainLoss of the previous timestep minus ElephantButteToCaballoLocalInflow.Local Inflow and the maximum flow at of the generator per current head is greater than the current calculated outflow.   
The rule then looks if Caballo.Pool Elevation in the previous timestep is less than the elevation of Animas in the CaballoData.CaballoSites table and if the current calculated release from Elephant Butte will cause Caballo to go back over the elevation of the Animas site and if the EBReleaseSchedule is equal to 1 or 3 (release to meet target or drawdown). If those criteria are true, then ElephantButte.Outflow is set to the maximum of the outflow calculated by energy input from the floor of the MWs interpolated from inflow needed to so that Caballo does not got over the Animas site and the minimum generator flow for the previous timestep?s head.

Model slots written by rule:  
1. ElephantButte.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ElephantButteData  
2. ElephantButte  
3. CaballoData  
4. Caballo  
5. ElephantButteToCaballoLocalInflow  
6. ElephantButteToCaballo

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
02/28/23: Lucas Barrett. Created Rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**MaxGeneratorFlowForHead
* **RPL Object Icon**MinGeneratorFlowForHead
* **RPL Object Icon**Min
* **RPL Object Icon**Ceiling
* **RPL Object Icon**Floor
* **RPL Object Icon**Max
* **RPL Object Icon**SolveInflow
* **RPL Object Icon**SolveOutflowGivenEnergyInflow
* **RPL Object Icon**ElevationToStorage
* **RPL Object Icon**FlowToVolume
* **RPL Object Icon**StorageToElevation
* **RPL Object Icon**TableInterpolation3D
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon3.8 RPL Object IconEB Release At Low Head

Rule Purpose:   
This rule only fires when EBPowerOptimization is turned on. When EB?s operational head is low (<~90 ft), two potential issue causing outcomes occur. First, generation, even with a maximum load on the generators, have limited MW production capabilities. Second, to keep at the same MW while head continues to decrease requires a significant amount more water. Due to these two outcomes, the model starts having issues maintaining a constant MW for a period of time while still aiming for a certain storage in Caballo. This rule tells the model that when the reservoir level falls below a specified operational head (LowestHeadToOptimizeGeneration, set to 90 ft as default), to keep outflows constant instead of MWs due to the volatility of flows when trying to keep a constant MW. The first time this rule fires, it will calculate a flow based on Caballo?s current storage, the storage target in CaballoData.StorageTargetsForPowerOptimization, and the sum of a specified amount of days (ElephantButte.DaysUntilEBCanChangeMW) forward in time in the CaballoData.TotalOutflowDirectlyInputToOverrideRules slot. This will calculate an average flow needed and will be set to Elephant Butte Outflow for a specified amount of days afterward (ElephantButte.DaysUntilEBCanChangeMW). After that specified period, a new flow will be calculated as long as the head is still below the LowestHeadToOptimizeGeneration.

Rule Logic:  
If Configuration.EBPowerOptimization = 1, Elephant Butte.Operating Head is less than ElephantButteData.LowestHeadToOptimizeGeneration (default 90ft), and ElephentButteData.EBReleaseSchedule does not equal 0 (Caballo not releasing), then Elephant Butte.Outflow is equal to either the average release calculated to keep Caballo at the specified storage target over a period of time ElephantButteData.DaysUntilEBCanChangeMW (default 14 days) or the previous day's ElephantButte.Outflow. This is determined by whether the previous day's ElephantButte.Outflow equals the ElephantButte.Outflow from the specified number of days in the past (i.e. ElephantButteData.DaysUntilEBCanChangeMW), or the previous day's ElephantButte.Storage is less than 2 \* ElephantButte.Minimum Storag,e or if half of the previous day's ElephantButte.Outflow is greater than the newly calculated average, and it is not on the weekend. If any of that criteria is not met, then ElephantButte.Outflow will equal the previous day's ElephantButte.Outflow.

Model slots written by rule:  
1. ElephantButte.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ElephantButteData  
2. ElephantButte  
3. CaballoData  
4. Caballo

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
7/30/20: Lucas Barrett. Created Rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**ElephantButteMinStorage
* **RPL Object Icon**CaballoStorageTargetEBPowerOptimization
* **RPL Object Icon**ElephantButteReleaseAtMinStorage
* **RPL Object Icon**IsWeekend
* **RPL Object Icon**Min
* **RPL Object Icon**SumSlot
* **RPL Object Icon**Max
* **RPL Object Icon**VolumeToFlow
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon3.9 RPL Object IconSet EB Release Override

Rule Purpose:  
If Elephant Butte outflow data are input in the ElephantButteData.TotalOutflowDirectlyInputToOverrideRules, then these data will be used instead of any data computed by rules, subject to physical constraints.

Rule Logic:  
This rule sets the value for the Outflow series slot on the ElephantButte reservoir object equal to the value in the ElephantButteData.TotalOutflowDirectlyInputToOverrideRules series slot. The outflow is constrained by the CheckThisResPhysicalConstraints function to be no less that the minimum required spill (there is no minimum required spill for Elephant Butte because its spillway is Regulated), to be no more than the maximum possible Elephant Butte outflow (given the inflow, the turbine release capacity, and the spillway capacity), and to not drawdown Elephant Butte below the ElephantButteData.Minimum Storage.  
The rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function, if the current timestep is after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay, and if a value is input for the current timestep on the ElephantButteData.TotalOutflowDirectlyInputToOverrideRules series slot.

Model slots written by rule:  
1. ElephantButte.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. ElephantButte  
3. ElephantButteData

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**CheckThisResPhysicalConstraints
* **RPL Object Icon**TotalOutflow
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon3.10 RPL Object IconSet EB Channel Capacity

Rule Purpose:  
If the Elephant Butte outflow is larger than the downstream channel capacity, then the Elephant Butte outflow is reset to equal the downstream channel capacity.

Rule Logic:  
If the value in the Outflow series slot on the ElephantButte reservoir object is larger than the value returned by the ElephantButteReleaseForDSChannelCapacity (which is equal to the value in the ElephantButteData.Channel Capacity scalar slot), then the ElephantButte Outflow is set equal to the value returned by the ElephantButteReleaseForDSChannelCapacity.  
The rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function, if the current timestep is after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay.

Model slots written by rule:  
1. ElephantButte.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. ElephantButte  
3. ElephantButteData

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**ElephantButteReleaseForDSChannelCapacity
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon3.11 RPL Object IconElephant Butte Spill Record

Rule Purpose:   
This rule determines if a conservation spill has occurred and makes a record of the spill. Note that in some cases the pool elevation of Elephant Butte reservoir may be over 4,414 feet but not officially spill for that timestep. This occurs if the release from Elephant Butte reservoir is required to meet the flood control constraint but not greater than the release from Caballo reservoir as required to meet downstream demands. The spill record is used by the Compact accounting rules to identify years in which spills occur.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the value for the EBSpill series slot on the Compact Data object with an IF THEN ELSE statement, which checks if the value in the Pool Elevation series slot on the ElephantButte reservoir object is larger than the value in the Flood Control Level scalar slot on the ElephantButteData object and if the minimum of the following is larger than the value in the Outflow series slot on the ElephantButte reservoir object:  
1. The maximum of the following:  
a. The value returned by the predefined SolveOutflow function, given the reservoir object for which to calculate is the ElephantButte reservoir object, the average inflow over the timestep in the value on the Inflow series slot on the ElephantButte reservoir object, the end of timestep storage is the value returned by the predefined function ElevationToStorage given the value in the Flood Control Level scalar slot on the ElephantButteData object, and the beginning of timestep storage is the value in the Storage series slot on the ElephantButte reservoir object on the previous timestep  
b. The value returned by the user-defined Increase Caballo Release to Meet Shortage function. The Increase Caballo Release to Meet Shortage function returns the Caballo Outflow plus 75% of the sum of the shortages at all of the LRG river headgates, plus a buffer of 10 acre-feet.  
2. The value in the ElephantButteData.Channel Capacity scalar slot.  
3. The maximum possible ElephantButte outflow (given the inflow, the turbine release capacity, and the spillway capacity).

If TRUE, the value for the EBSpill series slot on the Compact Data object is set equal to minimum of the following, minus the value in the Outflow series slot on the ElephantButte reservoir object:  
1. The maximum of the following:  
a. The value returned by the predefined SolveOutflow function, given the reservoir object for which to calculate is the ElephantButte reservoir object, the average inflow over the timestep in the value on the Inflow series slot on the ElephantButte reservoir object, the end of timestep storage is the value returned by the predefined function ElevationToStorage given the value in the Flood Control Level scalar slot on the ElephantButteData object, and the beginning of timestep storage is the value in the Storage series slot on the ElephantButte reservoir object on the previous timestep  
b. The value returned by the user-defined Increase Caballo Release to Meet Shortage function. The Increase Caballo Release to Meet Shortage function returns the Caballo Outflow plus 75% of the sum of the shortages at all of the LRG river headgates, plus a buffer of 10 acre-feet.  
2. The value in the ElephantButteData.Channel Capacity scalar slot.  
3. The maximum possible ElephantButte outflow (given the inflow, the turbine release capacity, and the spillway capacity).  
If FALSE, the value for the EBSpill series slot on the Compact Data object is set equal to 0 acre-feet/month.   
The rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. Compact Data.EBSpill

List of key model objects with slots read by the rule or child functions:  
1. ElephantButte  
2. ElephantButteData  
3. Caballo  
4. CaballoData

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**MinItem
* **RPL Object Icon**Max
* **RPL Object Icon**GetMaxOutflowGivenInflow
* **RPL Object Icon**ElevationToStorage
* **RPL Object Icon**SolveOutflow
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**Increase Caballo Release to Meet Shortage

### Control Display Icon3.12 RPL Object IconElephant Butte Flood Control

Rule Purpose:   
If the pool elevation of Elephant Butte reservoir is greater than the flood control level (4,407 feet) after making its Project releases, this rule increases the outflow so that the pool elevation drops to 4,182 feet.

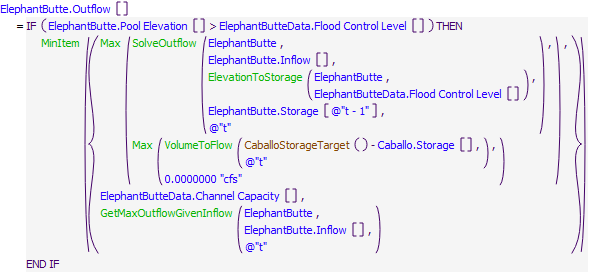
Rule Logic: Execution Constraint logic is at end of explanation.  
If the value in the Pool Elevation series slot on the ElephantButte reservoir object is more than the value in the Flood Control Level scalar slot on the ElephantButteData object, the value for the outflow series slot on the ElephantButte reservoir object is set equal to the minimum of:  
1. The value returned by the predefined SolveOutflow function, given the reservoir object for which to calculate is the ElephantButte reservoir object, the average inflow over the timestep in the value on the Inflow series slot on the ElephantButte reservoir object, the end of timestep storage is the value returned by the predefined function ElevationToStorage given the Flood Control Level, and the beginning of timestep storage is the value in the Storage series slot on the ElephantButte reservoir object on the previous timestep.   
2. The value in the Channel Capacity scalar slot on the ElephantButteData object.  
3. The maximum possible ElephantButte outflow (given the inflow, the turbine release capacity, and the spillway capacity).  
The rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function, if the value for the Pool Elevation series slot on the ElephantButte reservoir object is more than the value in the Flood Control Level scalar slot on the ElephantButteData object, and if the current timestep is after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay.

Model slots written by rule:  
1. ElephantButte.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. ElephantButte  
3. ElephantButteData  
4. Caballo  
5. CaballoData

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**CaballoStorageTarget
* **RPL Object Icon**MinItem
* **RPL Object Icon**Max
* **RPL Object Icon**GetMaxOutflowGivenInflow
* **RPL Object Icon**ElevationToStorage
* **RPL Object Icon**SolveOutflow
* **RPL Object Icon**VolumeToFlow
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon3.13 RPL Object IconSet Caballo Pattern Based Release

Rule Purpose:   
This rule sets the outflow from Caballo reservoir based on a release pattern (derived from historical releases), the Total Usable Water Available for Current Year Allocation, the hydrologic year type. The outflow is limited to the project storage, the maximum possible outflow (given the inflow, the release capacity, and the spillway capacity), and the physical storage at Caballo minus the minimum storage.

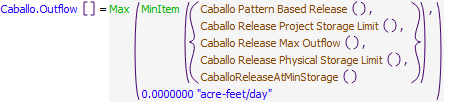
Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the value for the Outflow series slot on Caballo reservoir equal to the maximum of 0 acre-feet or the minimum of the values returned by the user-defined Caballo Pattern Based Release, Caballo Release Project Storage Limit, Caballo Release Max Outflow, and Caballo Release Physical Storage Limit functions.  
The Caballo Pattern Based Release function returns the Total Usable Water Available for Current Year Allocation multiplied by the value returned by an IF THEN ELSE statement which checks if the timestep is daily. If TRUE, the value returned is the value from the row corresponding to the current timestep and the GetYearTypeasString column in the CaballoReleasePattern periodic tableslot. If FALSE, the value returned is the value returned by the Monthly Pattern function for the CaballoReleasePattern periodic tableslot (which aggregates the daily release pattern coefficients to monthly).  
The Caballo Release Project Storage Limit function returns the sum of the Rio Grande Storage on Caballo and Elephant Butte, as a flow.   
The Caballo Release Max Outflow function returns the maximum possible Caballo outflow (given the inflow, the release capacity, and the spillway capacity).  
The Caballo Release Physical Storage Limit function returns the physical Caballo storage minus the CaballoData.Minimum Storage.  
This rule fires once per timestep if the following are all TRUE:  
1. If the Use Demand Based or Pattern Based Diversions switch on the Configuration object is set to 1.  
2. If the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function  
3. If the current timestep is after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay.  
4. If the RinconLocalTimestep is larger than the LRGDiversionStartDate for EBID OR if the MesillaLocalTimestep is larger than the LRGDiversionStartDate for EPCWID.  
5. If the Caballo Outflow is not already input.

Model slots written by rule:  
1. Caballo.Outflow

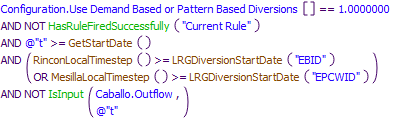
List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Caballo  
3. ElephantButte  
4. CaballoData  
5. Configuration  
6. CaballoToLeasburgTimeLag  
7. LeasburgToMesillaLag

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
1/28/21 : Marc Sidlow. Added function to allow minimum storage at Caballo (CaballoMinStorage)  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

****

Referenced Functions

* **RPL Object Icon**Caballo Pattern Based Release
* **RPL Object Icon**CaballoReleaseAtMinStorage
* **RPL Object Icon**Caballo Release Physical Storage Limit
* **RPL Object Icon**Caballo Release Project Storage Limit
* **RPL Object Icon**Caballo Release Max Outflow
* **RPL Object Icon**RinconLocalTimestep
* **RPL Object Icon**MesillaLocalTimestep
* **RPL Object Icon**LRGDiversionStartDate
* **RPL Object Icon**MinItem
* **RPL Object Icon**Max
* **RPL Object Icon**IsInput
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon3.14 RPL Object IconSet Caballo Release to Meet RHG Demands

Rule Purpose:   
This rule sets the outflow from Caballo reservoir based on the river headgate demands. This is an iterative process. The outflow from Caballo reservoir is increased each iteration until the shortage at American Diversion dam is zero, limited to the project storage and the maximum possible outflow (given the inflow, the release capacity, and the spillway capacity).

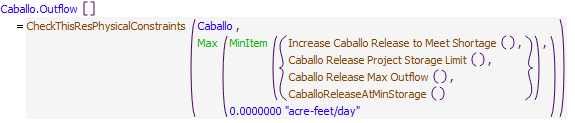
Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the value for the Outflow series slot on Caballo reservoir equal to the maximum of 0 acre-feet or the minimum of:   
1. The value returned by the Increase Caballo Release to Meet Shortage function, which is the Caballo Outflow plus 75% of the sum of the shortages at all of the LRG river headgates, plus a buffer of 10 acre-feet.  
2. The value returned by the Caballo Release Project Storage Limit function, which is the sum of the Rio Grande Storage on Caballo and Elephant Butte, as a flow.   
3. The value returned by the Caballo Release Max Outflow function, which is the maximum possible Caballo outflow (given the inflow, the release capacity, and the spillway capacity).  
The outflow is constrained by the CheckThisResPhysicalConstraints function to be no less that the minimum required spill (there is no minimum required spill for Caballo because its spillway is Regulated), to be no more than the maximum possible Caballo outflow, and to not drawdown Caballo below the CaballoData.Minimum Storage.  
This rule fires once per timestep if the following are all TRUE:  
1. If the River Head Gate Shortage is more than 0.1 acre-feet/day.  
2. If it is Caballo Release Season.  
3. If the current timestep is after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay.  
4. If the Use Demand Based or Pattern Based Diversions switch on the Configuration object is set to 0.  
5. If the RinconLocalTimestep is larger than the LRGDiversionStartDate for EBID OR if the MesillaLocalTimestep is larger than the LRGDiversionStartDate for EPCWID.  
6. If the Caballo Outflow is not already input.

Model slots written by rule:  
1. Caballo.Outflow

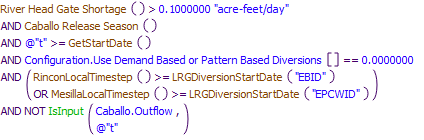
List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Caballo  
3. ElephantButte  
4. CaballoData  
5. Configuration  
6. D3 Data  
7. CaballoToLeasburgTimeLag  
8. LeasburgToMesillaLag

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
1/28/21 : Marc Sidlow. Added function to allow minimum storage at Caballo (CaballoMinStorage)  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

****

Referenced Functions

* **RPL Object Icon**Caballo Release Season
* **RPL Object Icon**CheckThisResPhysicalConstraints
* **RPL Object Icon**CaballoReleaseAtMinStorage
* **RPL Object Icon**Caballo Release Project Storage Limit
* **RPL Object Icon**Caballo Release Max Outflow
* **RPL Object Icon**RinconLocalTimestep
* **RPL Object Icon**MesillaLocalTimestep
* **RPL Object Icon**LRGDiversionStartDate
* **RPL Object Icon**MinItem
* **RPL Object Icon**Max
* **RPL Object Icon**IsInput
* **RPL Object Icon**GetStartDate
* **RPL Object Icon**River Head Gate Shortage
* **RPL Object Icon**Increase Caballo Release to Meet Shortage

### Control Display Icon3.15 RPL Object IconSet Caballo Release Override

Rule Purpose:  
If Caballo outflow data are input in the CaballoData.TotalOutflowDirectlyInputToOverrideRules, then these data will be used instead of any data computed by rules, subject to physical constraints.

Rule Logic:  
This rule sets the value for the Outflow series slot on the Caballo reservoir object equal to the value in the CaballoData.TotalOutflowDirectlyInputToOverrideRules series slot. The outflow is constrained by the CheckThisResPhysicalConstraints function to be no less that the minimum required spill (there is no minimum required spill for Caballo because its spillway is Regulated), to be no more than the maximum possible Caballo outflow (given the inflow, the release capacity, and the spillway capacity), and to not drawdown Caballo below the CaballoData.Minimum Storage.  
The rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function, if the current timestep is after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay, if a value is input for the current timestep on the CaballoData.TotalOutflowDirectlyInputToOverrideRules series slot, and if Caballo Outflow is not already input.

Model slots written by rule:  
1. Caballo.Outflow

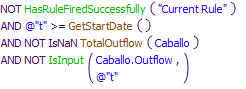
List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Caballo  
3. CaballoData

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

**Statements**

Execution Constraint

****

Referenced Functions

* **RPL Object Icon**CheckThisResPhysicalConstraints
* **RPL Object Icon**TotalOutflow
* **RPL Object Icon**CaballoReleaseAtMinStorage
* **RPL Object Icon**Min
* **RPL Object Icon**IsInput
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon3.16 RPL Object IconCaballoDSChannelCapacityRule

Rule Purpose:  
The maximum outflow from Cochiti Dam to maintain flows at San Marcial below the channel capacity at San Marcial is determined and recorded with this rule. The channel capacity at San Marcial is checked as input but will also be reduced if needed such that Elephant Butte Reservoir is not filled resulting in channel capacities below Elephant Butte from being exceeded.

Rule Logic:  
This rule sets the value in the MaxReleaseForDSChannelCap series slot on the CaballoData object equal to the value in the Channel Capacity series slot on the CaballoData object.  
This rule fires if the current timestep is after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay, and if MaxReleaseForDSChannelCap is not already input.

Model slots written by rule:  
1. CaballoData.MaxReleaseForDSChannelCap

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. CaballoData

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**GetStartDate

### Control Display Icon3.17 RPL Object IconElPasoChannelCapacityRule

Rule Purpose:  
The maximum outflow from Cochiti Dam to maintain flows at San Marcial below the channel capacity at San Marcial is determined and recorded with this rule. The channel capacity at San Marcial is checked as input but will also be reduced if needed such that Elephant Butte Reservoir is not filled resulting in channel capacities below Elephant Butte from being exceeded.

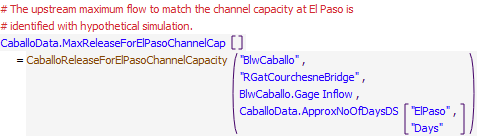
Rule Logic:  
This rules sets the value in the MaxReleaseForDSChannelCap series slot on the CaballoData object equal to the value returned by the CaballoReleaseForElPasoChannelCapacity function, which performs a Hypothetical Simulation given that the subbasin is BlwCaballoToRGatCourchesneBridge, the control slot is BlwCaballo.Gage Inflow, the minimum control slot value is 0 cfs, the maximum control slot value is 17,000 cfs, the target slot is RGatCourchesneBridge.Gage Inflow, the target timestep is the timestep at Caballo plus the value in the ElPaso column in the CaballoData.ApproxNoOfDaysDS tableslot, the target value is the value in the Capacity row and ElPaso column of the CaballoData.ChannelCapacities tableslot, the target tolerance is the value in the Tolerance row and ElPaso column of the CaballoData.ChannelCapacities tableslot, the maximum number of iterations is 20, and the minimum number of timesteps before and after the current timestep which might be involved in the simulation is 0.  
This rule fires if the current timestep is after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay, and if MaxReleaseForElPasoChannelCap is not already input.

Model slots written by rule:  
1. CaballoData.MaxReleaseForElPasoChannelCap

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. BlwCaballo  
3. CaballoData

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**CaballoReleaseForElPasoChannelCapacity
* **RPL Object Icon**GetStartDate

### Control Display Icon3.18 RPL Object IconCaballoOutflowRestrictions

Rule Purpose:   
If the Caballo outflow is larger the downstream channel capacity, or the resulting flow at El Paso is larger than the El Paso channel capacity, then the Caballo outflow is reduced so these capacities are not exceeded.

Rule Logic:  
This rule sets the value for the Outflow series slot on the Caballo reservoir object equal to the value in returned by the CaballoReleaseForChannelCapacity function, which is the minimum of the values in the MaxReleaseForDSChannelCap and MaxReleaseForElPasoChannelCap series slots on the CaballoData object.  
The outflow is constrained by the CheckThisResPhysicalConstraints function to be no less that the minimum required spill (there is no minimum required spill for Caballo because its spillway is Regulated), to be no more than the maximum possible Caballo outflow (given the inflow, the release capacity, and the spillway capacity), and to not drawdown Caballo below the CaballoData.Minimum Storage.  
The rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function, if the current timestep is after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay, if Caballo Outflow is larger than the maximum of the MaxReleaseForDSChannelCap and MaxReleaseForElPasoChannelCap, and if Caballo Outflow is not already input.

Model slots written by rule:  
1. Caballo.Outflow

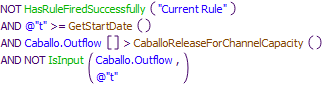
List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Caballo  
3. CaballoData

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

**Statements**

Execution Constraint

****

Referenced Functions

* **RPL Object Icon**CheckThisResPhysicalConstraints
* **RPL Object Icon**CaballoReleaseForChannelCapacity
* **RPL Object Icon**IsInput
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon3.19 RPL Object IconCaballo Flood Control

Rule Purpose:   
If the pool elevation of Caballo reservoir is greater than the flood control level (4,182 feet) after making its Project releases, this rule increases the outflow so that the pool elevation drops to 4,182 feet. If this is the case, this rule also sets the CaballoSpillFlag to 1, which is considered in subsequent Diversion Ratio computations.

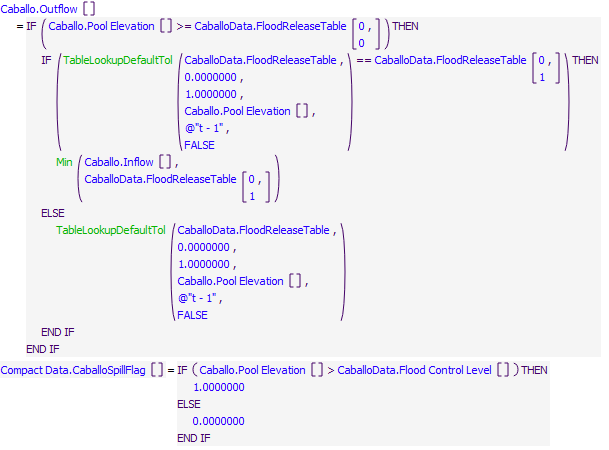
Rule Logic: Execution Constraint logic is at end of explanation.  
This rule has two assignments. In the first assignment, if the value in the Pool Elevation series slot on the Caballo reservoir object is more than the value in the Flood Control Level scalar slot on the CaballoData object, the value for the outflow series slot on the Caballo reservoir object is set equal to the minimum of:  
1. The value returned by the predefined SolveOutflow function, given the reservoir object for which to calculate is the Caballo reservoir object, the average inflow over the timestep in the value on the Inflow series slot on the Caballo reservoir object, the end of timestep storage is the value returned by the predefined function ElevationToStorage given the Flood Control Level, and the beginning of timestep storage is the value in the Storage series slot on the Caballo reservoir object on the previous timestep.   
2. The value in the Channel Capacity scalar slot on the CaballoData object.  
3. The maximum possible Caballo outflow (given the inflow, the release capacity, and the spillway capacity).  
In the second assignment, if the value in the Pool Elevation series slot on the Caballo reservoir object is more than the value in the Flood Control Level scalar slot on the CaballoData object, the value for the CaballoSpillFlag on the Compact Data object is set equal to 1, and if not, the value for the CaballoSpillFlag on the Compact Data object is set equal to 0.   
The rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function, if the value for the Pool Elevation series slot on the Caballo reservoir object is more than the value in the Flood Control Level scalar slot on the CaballoData object, if the current timestep is after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay, and if Caballo Outflow is not already input.

Model slots written by rule:  
1. Caballo.Outflow  
2. Compact Data.CaballoSpillFlag

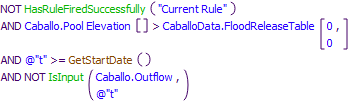
List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Caballo  
3. CaballoData

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

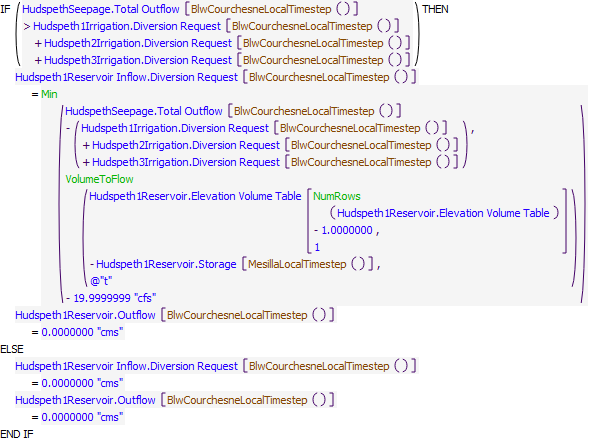
****

Referenced Functions

* **RPL Object Icon**Min
* **RPL Object Icon**TableLookupDefaultTol
* **RPL Object Icon**IsInput
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon3.20 RPL Object IconFill Hudspeth Reservoir 1

Statements

****

Execution Constraint

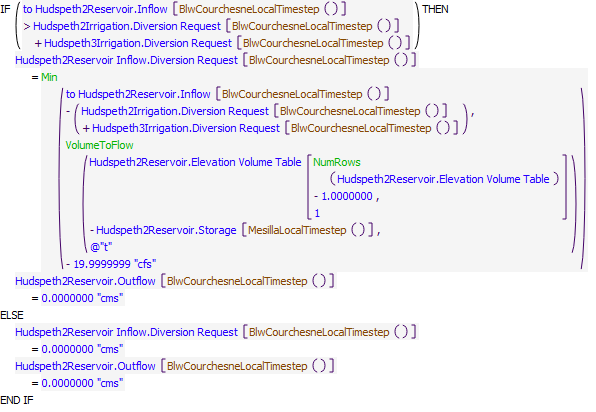
**Execution Constraint**

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**MesillaLocalTimestep
* **RPL Object Icon**Min
* **RPL Object Icon**VolumeToFlow
* **RPL Object Icon**NumRows
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon3.21 RPL Object IconFill Hudspeth Reservoir 2

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**MesillaLocalTimestep
* **RPL Object Icon**Min
* **RPL Object Icon**VolumeToFlow
* **RPL Object Icon**NumRows
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon3.22 RPL Object IconOperate Hudspeth Reservoir 3

Statements

****

Execution Constraint

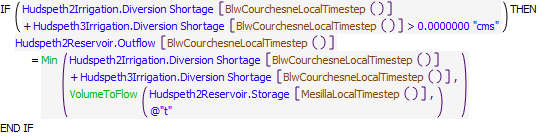
**Execution Constraint**

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**MesillaLocalTimestep
* **RPL Object Icon**Min
* **RPL Object Icon**Max
* **RPL Object Icon**VolumeToFlow
* **RPL Object Icon**NumRows
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon3.23 RPL Object IconRelease From Hudspeth Reservoir 2

Statements

****

Execution Constraint

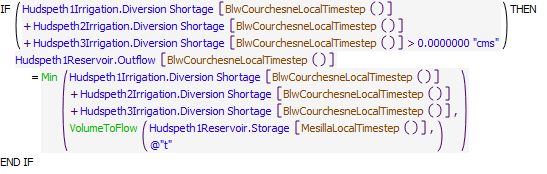
**Execution Constraint**

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**MesillaLocalTimestep
* **RPL Object Icon**Min
* **RPL Object Icon**VolumeToFlow
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon3.24 RPL Object IconRelease From Hudspeth Reservoir 1

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**MesillaLocalTimestep
* **RPL Object Icon**Min
* **RPL Object Icon**VolumeToFlow
* **RPL Object Icon**HasRuleFiredSuccessfully

## Control Display Icon4 RPL Object IconLRG Project & Compact Accounting

This policy group is used to charge water users (EBID, EPCWID, Mexico) for their deliveries from Caballo. The charges are made against each water user's account on Caballo. Other rules set the release of Rio Grande water from Elephant Butte reservoir, and transfer relinquished NM Credit water, or Albuquerque or Santa Fe City payback water to the Rio Grande account.

### Control Display Icon4.1 RPL Object IconSetEBAccountingSupplyForRelinquishment

Rule Purpose:  
If relinquished Compact credits are being modeled and a relinquishment for the current timestep has been determined, or input, the accounting supply is set with this rule to transfer the relinquishment water from the NMCredit account to the Rio Grande account at Elephant Butte Reservoir.

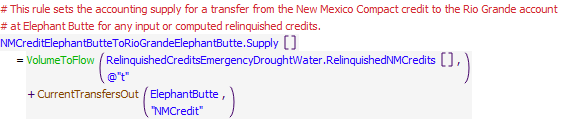
Rule Logic: Execution Constraint logic is at end of explanation.  
The NMCreditElephantButteToRioGrandeElephantButte.Supply is set equal to the value in the RelinquishedCreditsEmergencyDroughtWater.RelinquishedNMCredits series slot, converted to a flow, plus the value in the ElephantButte^NMCredit.Transfers Out series slot.  
This rule fires every timestep if the current timestep is after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay, if RelinquishedCreditsEmergencyDroughtWater.TriggerModelRelinquishedCredits switch is non-zero, if the current date is equal to the value in the DateOfRelinquishment column of the RelinquishedCreditsEmergencyDroughtWater.RelinquishedCreditsTriggers tableslot, and if the value in the RelinquishedCreditsEmergencyDroughtWater.RelinquishedNMCredits series slot is more than 0 acre-feet.

Model slots written by rule:  
1. NMCreditElephantButteToRioGrandeElephantButte.Supply

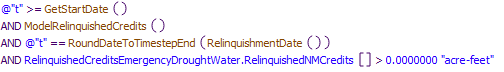
List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. RelinquishedCreditsEmergencyDroughtWater  
3. ElephantButte

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

****

Referenced Functions

* **RPL Object Icon**RoundDateToTimestepEnd
* **RPL Object Icon**RelinquishmentDate
* **RPL Object Icon**ModelRelinquishedCredits
* **RPL Object Icon**VolumeToFlow
* **RPL Object Icon**CurrentTransfersOut
* **RPL Object Icon**GetStartDate

### Control Display Icon4.2 RPL Object IconSetElephantButteAccounts

Rule Purpose:   
This rule sets the accounting supply for the release of Rio Grande water out of Elephant Butte Reservoir to the total outflow. Accounting supplies are also set in this rule for transfers of San Juan-Chama Project water to the Rio Grande account for contractors that have a payback. Albuquerque and Santa Fe may have a pending payback for surface water diversions conducted during Abiquiu flood control operations when native water was diverted to be paid back with San Juan-Chama Project water at Elephant Butte.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule has three assignments.   
The first assignment sets the RioGrandeElephantButteToRioGrandeBlwElephantButte.Supply equal to the ElephantButte.Outflow.

The second assignment sets the AlbuquerqueElephantButteToRioGrandeElephantButte.Supply with an IF THEN ELSE statement which checks if the current timestep is equal to the date in the Albuquerque row and DateForTransferToRioGrandeAtElephantButteForPayback column of the ComputedDeliveries.DeliverySettings object.  
If FALSE, the AlbuquerqueElephantButteToRioGrandeElephantButte.Supply is set to 0 cfs.  
If TRUE, the AlbuquerqueElephantButteToRioGrandeElephantButte.Supply is set to the maximum of 0 cfs or the minimum of the value returned by the GetPaybackDebt function, given the payback source is the AlbuquerqueElephantButteToRioGrandeElephantButte.Supply, or the previous ElephantButte^Albuquerque storage + the previous ElephantButte^Albuquerque gain loss.

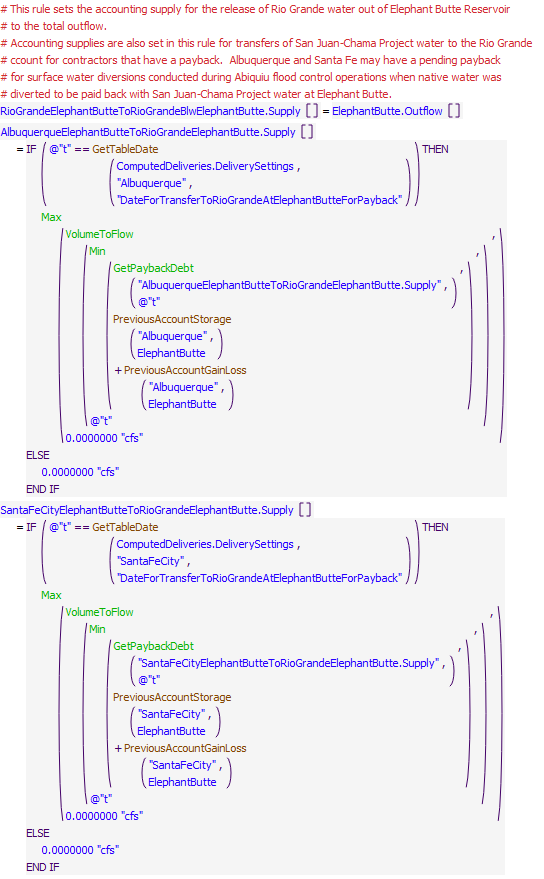
The third assignment sets the SantaFeCityElephantButteToRioGrandeElephantButte.Supply with an IF THEN ELSE statement which checks if the current timestep is equal to the date in the SantaFeCity row and DateForTransferToRioGrandeAtElephantButteForPayback column of the ComputedDeliveries.DeliverySettings object.  
If FALSE, the SantaFeCityElephantButteToRioGrandeElephantButte.Supply is set to 0 cfs.  
If TRUE, the SantaFeCityElephantButteToRioGrandeElephantButte.Supply is set to the maximum of 0 cfs or the minimum of the value returned by the GetPaybackDebt function, given the payback source is the SantaFeCityElephantButteToRioGrandeElephantButte.Supply, or the previous ElephantButte^SantaFeCity storage + the previous ElephantButte^SantaFeCity gain loss.   
The rule fires once per timestep if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function, and if the current timestep is after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay.

Model slots written by rule:  
1. RioGrandeElephantButteToRioGrandeBlwElephantButte.Supply  
2. AlbuquerqueElephantButteToRioGrandeElephantButte.Supply  
3. SantaFeCityElephantButteToRioGrandeElephantButte.Supply

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. ElephantButte  
3. ComputedDeliveries

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**GetTableDate
* **RPL Object Icon**Min
* **RPL Object Icon**Max
* **RPL Object Icon**GetPaybackDebt
* **RPL Object Icon**VolumeToFlow
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**PreviousAccountGainLoss
* **RPL Object Icon**GetStartDate
* **RPL Object Icon**PreviousAccountStorage

### Control Display Icon4.3 RPL Object IconMexico Diversions

Rule Purpose:   
The Mexico account on Caballo gets charged the amount of water diverted at the Acequia Madre headgate.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the value for the Outflow series slot on the Mexico account on the Caballo reservoir object to the value on the Diversion series slot on the AcequiaMadre diversion object at the BlwCourchesneLocalTimestep.  
This rule fires every timestep.

Model slots written by rule:  
1. Caballo^Mexico.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. AcequiaMadre  
3. CaballoToLeasburgTimeLag  
4. LeasburgToMesillaLag  
5. MesillaToAmericanLag

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

**Statements**

Referenced Functions

* **RPL Object Icon**BlwCourchesneLocalTimestep

### Control Display Icon4.4 RPL Object IconEBID Deliveries

Rule Purpose:   
The EBID account on Caballo gets charged the sum of water diverted at the Rincon, Leasburg, Mesilla East to New Mexico and Mesilla West to New Mexico headgates. If EPCWID is not diverting, and EBID is, then the EBID account gets charged the full Caballo outflow minus the Acequia Madre diversion.

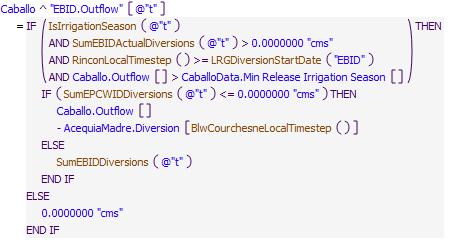
Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the value for the Outflow series slot on the EBID account on the Caballo reservoir object with an IF THEN ELSE statement, which checks if the current timestep is within the irrigation season, which is February through October, with reference to the user-defined IsIrrigationSeason function, if the value returned by the SumEBIDActualDiversions function is more than 0 acre-feet/month, if the RinconLocalTimestep is larger than the LRGDiversionStartDate for EBID, and if the Caballo Outflow is larger than the value in the CaballoData.Min Release Irrigation Season scalar slot.  
If FALSE, the value for the Outflow series slot on the EBID account on the Caballo reservoir object is set to 0 acre-feet/month.   
If TRUE, an internal IF THEN ELSE statement checks if the value returned by the SumEPCWIDDiversions function is less than or equal to 0 acre-feet/month.  
If TRUE, the value for the Outflow series slot on the EBID account on the Caballo reservoir object is set to the current Caballo Outflow minus the BlwCourchesneLocalTimestep value on the Diversion Slot on the Acequia Madre diversion object.  
If FALSE, the value for the Outflow series slot on the EBID account on the Caballo reservoir object is set to the value returned by the user-defined SumEBIDDiversions function.   
The SumEPCWIDDiversions function sums the values in the Eastside Canal at State Line.Gage Outflow, La Union Canal at State Line.Gage Outflow, American Diversion.Diversion, and Riverside Coffer Dam.Diversion series slots.  
The SumEBIDDiversions function is the sum of the values in the ArreyandPercha.Diversion and LeasburgandPicacho.Diversion series slots, plus the value in the Eastside Diversion slot minus 1.5 times the value in the Eastside Canal at State Line.Gage Outflow slot, limited to be no less than 0 acre-feet/day, plus the value in the Westside Diversion slot minus 1.5 times the value in the La Union Canal at State Line.Gage Outflow slot, limited to be no less than 0 acre-feet/day.  
This rule fires every timestep.

Model slots written by rule:  
1. Caballo^EBID.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Eastside Canal at State Line  
3. La Union Canal at State Line  
4. American Diversion  
5. Riverside Coffer Dam  
6. CaballoToLeasburgTimeLag  
7. LeasburgToMesillaLag  
8. MesillaToAmericanLag  
9. Caballo  
10. CaballoData  
11. ArreyandPercha  
12. LeasburgandPicacho  
13. Rio Grande Project Data  
14. AcequiaMadre  
15. Eastside  
16. Westside

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Referenced Functions

* **RPL Object Icon**SumEBIDActualDiversions
* **RPL Object Icon**SumEPCWIDDiversions
* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**RinconLocalTimestep
* **RPL Object Icon**LRGDiversionStartDate
* **RPL Object Icon**IsIrrigationSeason
* **RPL Object Icon**SumEBIDDiversions

### Control Display Icon4.5 RPL Object IconEPCWID Deliveries

Rule Purpose:  
The EPCWID account on Caballo gets charged the sum of water diverted at the Mesilla Eastside to Texas, Mesilla Westside to Texas, and American Diversion headgates. Also includes Haskell Street credits for M&I. If EBID is not diverting, and EPCWID is, then the EPCWID account gets charged the full Caballo outflow minus the Acequia Madre diversion.

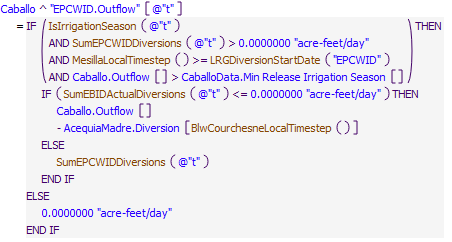
Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the value for the Outflow series slot on the EPCWID account on the Caballo reservoir object with an IF THEN ELSE statement, which checks if the current timestep is within the irrigation season, which is February through October, with reference to the user-defined IsIrrigationSeasonfunction, if the value returned by the SumEPCWIDActualDiversions function is more than 0 acre-feet/month, if the MesillaLocalTimestep is larger than the LRGDiversionStartDate for EPCWID and if the Caballo Outflow is larger than the value in the CaballoData.Min Release Irrigation Season scalar slot.  
If FALSE, the value for the Outflow series slot on the EPCWID account on the Caballo reservoir object is set to 0 acre-feet/month.   
If TRUE, an internal IF THEN ELSE statement checks if the value returned by the SumEBIDDiversions function is less than or equal to 0 acre-feet/month.  
If TRUE, the value for the Outflow series slot on the EPCWID account on the Caballo reservoir object is set to the current Caballo Outflow minus the BlwCourchesneLocalTimestep value on the Diversion Slot on the Acequia Madre diversion object.  
If FALSE, the value for the Outflow series slot on the EPCWID account on the Caballo reservoir object is set to the value returned by the user-defined SumEPCWIDDiversions function.   
The SumEPCWIDDiversions function sums the values in the Eastside Canal at State Line.Gage Outflow, La Union Canal at State Line.Gage Outflow, American Diversion.Diversion, and Riverside Coffer Dam.Diversion series slots.  
The SumEBIDDiversions function is the sum of the values in the ArreyandPercha.Diversion and LeasburgandPicacho.Diversion series slots, plus the value in the Eastside Diversion slot minus 1.5 times the value in the Eastside Canal at State Line.Gage Outflow slot, limited to be no less than 0 acre-feet/day, plus the value in the Westside Diversion slot minus 1.5 times the value in the La Union Canal at State Line.Gage Outflow slot, limited to be no less than 0 acre-feet/day.  
This rule fires every timestep.

Model slots written by rule:  
1. Caballo^EPCWID.Outflow

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Eastside Canal at State Line  
3. La Union Canal at State Line  
4. American Diversion  
5. Riverside Coffer Dam  
6. CaballoToLeasburgTimeLag  
7. LeasburgToMesillaLag  
8. Caballo  
9. CaballoData  
10. ArreyandPercha  
11. LeasburgandPicacho  
12. Rio Grande Project Data  
13. AcequiaMadre

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
9/4/18: Nick Mander. Added description field.   
3/23/15: Nick Mander. Created the rule.

Statements

****

Referenced Functions

* **RPL Object Icon**SumEBIDActualDiversions
* **RPL Object Icon**SumEPCWIDDiversions
* **RPL Object Icon**BlwCourchesneLocalTimestep
* **RPL Object Icon**MesillaLocalTimestep
* **RPL Object Icon**LRGDiversionStartDate
* **RPL Object Icon**IsIrrigationSeason