# Appendix F: Individual Colorado 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 Colorado 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 Colorado 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.

## Control Display Icon1 RPL Object IconSet Colorado Diversion Requests

This policy group sets initial diversion requests in Colorado. These requests logically fit into the Policy Group "Set Colorado Diversions", however the initial diversion requests are used in Platoro operations, so are in a separate group here. Two of the rules in this policy group fire at the beginning of the model run to set the initial requests in all Rio Grande and Conejos diversions to the full water right of each diversion. The third rule modifies initial requests along the Conejos based on an estimate of saturated land demand conditions to allow senior users whose needs have been fully met to store that water in Platoro in lieu of diversion under certain conditions.

Policy Group Change Log (newest changes at the top):   
Date: Who. What  
8/13/2018: Jesse Roach. Added Group Description and Group Notes fields.  
5/5/2015: Jesse Roach. Added the rule ModifiyConejosInitialRequests to this Policy Group.   
~2013: Craig Boroughs Created this Policy Group and the SetConejosInitialRequests and SetRioGrandeInitialRequests rules.

### Control Display Icon1.1 RPL Object IconSetConejosInitialRequests

Rule Purpose:  
This rule sets the "Initial Request" slots on all diversion accounts along the Conejos, Los Pinos, and San Antonio Rivers in Colorado for all time steps. Initial requests for timesteps within the irrigation season (April 1 - October 31) are set based on water right. Outside of irrigation season, the initial request is set to zero. This rule only executes once, on the first time step of the simulation.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule loops through all diversion objects along the Conejos, Los Pinos, and San Antonio Rivers in Colorado, and for each water owner sets the 'Initial Request' on the diversion accout for all timesteps in the model run. Irrigation season diversion requests are set based on the total water right by reading water right owner specific values from the InitialRequestValues.RioGrandeDistrict20WaterRights table slot. Non-irrigation season diversion requests are set to zero.

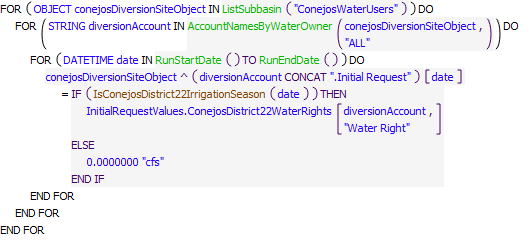
This rules fires at first timestep of the simulation

Model slots written by rule:  
1. 'Water User Object'^Account.Initial Request for all Water User Objects diverting from the Conejos, Los Pinos, or San Antonio river in Colorado. This is a total of 190 slots on the Conejos system.

List of key model objects with slots read by the rule or child functions:  
1. InitialRequestValues (Data Object)

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
~2013: Craig Boroughs Created the rule ? slightly modified in 2015 by unknown when MRG rules were merged to include Colorado portion.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**IsConejosDistrict22IrrigationSeason
* **RPL Object Icon**RunStartDate
* **RPL Object Icon**RunEndDate
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**AccountNamesByWaterOwner

### Control Display Icon1.2 RPL Object IconSetRioGrandeInitialRequests

Rule Purpose:  
This rule sets the "Initial Request" slots on all diversion accounts along the Rio Grande for all time steps. Initial requests for timesteps within the irrigation season (April 1 - October 31) are set based on water right. Outside of irrigation season, the initial request is set to zero. This rule only executes once, on the first time step of the simulation.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule loops through all diversion objects along the Rio Grande in Colorado, and for each water owner sets the 'Initial Request' on the diversion account for all timesteps in the model run. Irrigation season diversion requests are set based on the total water right by reading water right owner specific values from the InitialRequestValues.RioGrandeDistrict20WaterRights table slot. Non-irrigation season diversion requests are set to zero.

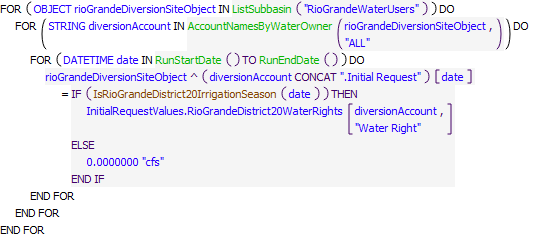
This rules fires at first timestep of the simulation

Model slots written by rule:  
1. 'Water User Object'^Account.Initial Request for all Water User Objects diverting from the Rio Grande in Colorado. This is a total of 298 slots on the Rio Grande in Colorado.

List of key model objects with slots read by the rule or child functions:  
1. InitialRequestValues (Data Object)

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
~2013: Craig Boroughs Created the rule ? slightly modified in 2015 by unknown when MRG rules were merged to include Colorado portion.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**IsRioGrandeDistrict20IrrigationSeason
* **RPL Object Icon**RunStartDate
* **RPL Object Icon**RunEndDate
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**AccountNamesByWaterOwner

### Control Display Icon1.3 RPL Object IconModifyConejosInitialRequests

Rule Purpose:  
This rule reduces the initial request of water users along the Conejos River whose demand has been met for a given number of days consecutively. This is to simulate reduced demand due to saturated lands leading to Direct Flow Storage in Platoro. Direct Flow Storage in Platoro which occurs when a user on the Conejos system decides to store water in Platoro that it would otherwise be in priority to divert, for use later in the same irrigation season. See the main body of the URGWOM Rules Documentation for more information on Direct Flow Storage at Platoro.

Rule Logic: Execution Constraint logic is at end of explanation.  
If Direct Flow Storage can occur (irrigation season, and within a certain period of time after peak inflow), this rule loops through all Conejos Water User Objects and evaluates how many consecutive timesteps the full request or reduced request of each water right has been met. If the request was met in the previous timestep, and the consecutive timesteps full requests have been met exceeds a threshold value, or the consecutive timesteps partial requests have been met is less than a threshold, the initial request is reduced. These two cases test for the beginning and end of reduced requests respectively. Anytime a full or reduced request is not fully met, the consecutive timesteps counters reset. Thus reduced requests only begin after a threshold amount of time with fully met demands has passed, and the reduced requests continue as long as they are fully met for a different specified amount of time.

This rule fires if the current timestep is greater than or equal to the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND If the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. 'Water User Object'^Account.Initial Request for all Water User Objects diverting from the Conejos river in Colorado.

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. PlatoroData (Data Object)  
3. InitialRequestValues (Data Object)

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
5/5/2015: Jesse Roach. Created rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**RoundDurationToTimestep
* **RPL Object Icon**FullyServed
* **RPL Object Icon**ReducedDemandTargetDuration
* **RPL Object Icon**FullDemandTargetDuration
* **RPL Object Icon**ConsecutiveReducedRequests
* **RPL Object Icon**ConsecutiveFullRequests
* **RPL Object Icon**IsPlatoroDirectFlowStorageSeason
* **RPL Object Icon**GetTimestep
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**AccountNamesByWaterOwner
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

## Control Display Icon2 RPL Object IconPlatoro Conservation Storage Policy

This policy group implements Platoro conservation storage, which is storage in Platoro for reasons not related to flood control. Conservation storage occurs in Platoro under three different policy mechanisms, 1) Decree Storage, 2) Compact Storage, and 3) Direct Flow Storage. URGWOM simulates Decree Storage and Direct Flow Storage. Because URGWOM is able to curtail irrigation diversions in the Conejos sufficiently to make accurate compact deliveries, and because Compact storage in Platoro has been small historically, it is not included in URGWOM.

Policy Group Change Log (newest changes at the top):   
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.   
8/13/2018: Jesse Roach. Added Group Description and Group Notes fields.  
5/7/2015: Jesse Roach. Modified policy group significantly. Grouped existing conservation storage related rules and added new or modified rules associated with CWCD demand and relinquished credit accounting.

### Control Display Icon2.1 RPL Object IconComputeCWCDReleaseTargetAsFractionMaxConservationStorage

Rule Purpose:  
This rule calculates a target release of Conejos Water Conservancy District Decree Storage water. This release target is expressed as a fraction of peak conservation storage at Platoro in the current year, and is found as a function of total inflows to Platoro in the current year. Decree Water releases in a given year as a function of maximum conservation storage are determined based on an empirical linear relationship between annual Platoro inflows and the percentage of max storage released between peak storage and November 1st, developed by analysis of historical operations from 1993 through 2001. For Platoro inflows of 40,000 AF (dry), the model targets a release of 56% of peak storage, and for inflows of 100,000 AF (wet), targets a release of 20% of peak storage.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sums all inflows to Platoro for the current year, and uses that value to interpolate into the table PlatoroData.CWCDReleaseParameters to find the release target as a fraction of maximum conservation storage. The release target is written to PlatoroData.CWCDReleaseTargetAsFractionMaxConservationStorage, and referenced later by the rule ComputePlatoroCWCDDemand.

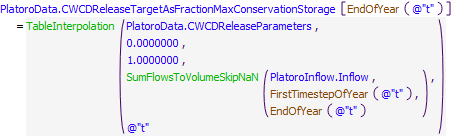
This rule fires if has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND the timestep is either the first timestep of the year OR is the start of the model run.

Model slots written by rule:  
1. PlatoroData.CWCDReleaseTargetAsFractionMaxConservationStorage

List of key model objects with slots read by the rule or child functions:  
1. PlatoroData (Data Object)  
2. PlatoroInflow (Reach Object)

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/18/2018: Jesse Roach. Updated execution constraint so that rule only fires on the first timestep of the year or start of the run, and only if it hasn't fired successfully already.  
5/25/2017: Jesse Roach. Created rule

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**EndOfYear
* **RPL Object Icon**IsStartOfRun
* **RPL Object Icon**FirstTimestepOfYear
* **RPL Object Icon**IsFirstTimestepOfYear
* **RPL Object Icon**SumFlowsToVolumeSkipNaN
* **RPL Object Icon**TableInterpolation
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon2.2 RPL Object IconSetPlatoroMaxAnnualInflowDate

Rule Purpose:  
The SetPlatoroMaxAnnualInflowDate rule evaluates the annual inflows to Platoro for the current year and returns the date of the peak inflow. This date is used as a flag for the onset of direct flow storage in in Platoro. This rule only works if the inflow for the current year is known ahead of time. If peak inflow were to be estimated based on antecedant data only, this rule would need to be reworked.

Rule Logic: Execution Constraint logic is at end of explanation.  
The SetPlatoroMaxAnnualInflowDate rule sorts all PlatoroInflow.Inflow data for the current year by size and returns the date associated with the first (largest) flow.

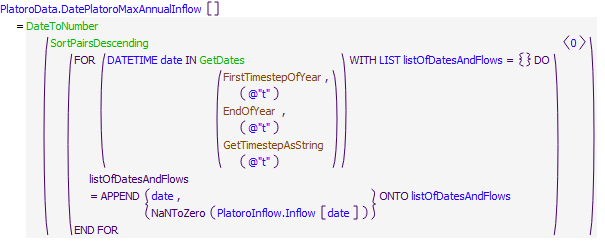
This rule fires if has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND the timestep is either the first timestep of the year OR is the start of the model run.

Model slots written by rule:  
1. PlatoroData.DatePlatoroMaxAnnualInflow

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
11/17/2017: Jesse Roach. Modified rule during timestep generalization.  
5/4/2015: Jesse Roach. Created rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**EndOfYear
* **RPL Object Icon**IsStartOfRun
* **RPL Object Icon**GetTimestepAsString
* **RPL Object Icon**FirstTimestepOfYear
* **RPL Object Icon**IsFirstTimestepOfYear
* **RPL Object Icon**DateToNumber
* **RPL Object Icon**GetDates
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**SortPairsDescending

### Control Display Icon2.3 RPL Object IconComputeReducedDiversionRequestsBelowPlatoro

Rule Purpose:  
If irrigation demand below Platoro is being reduced, and it is irrigation season on the Conejos, the ComputeReducedDiversionRequestsBelowPlatoro rule records the reduced diversion requests downstream of Platoro. Depending on policy, these reduced requests, or some portion of them could be stored in Platoro in lieu of diversion for use later in the same season. Reduced diversion requests may result when a diverter has had full demand served for a certain number of days. The reduced diversion requests are calculated for each diversion account in the rule ModifyConejosInitialRequests.

Rule Logic: Execution Constraint logic is at end of explanation.  
If it is irrigation season and irrigation demand is being reduced for the purposes of direct flow storage, the ComputeReducedDiversionRequestsBelowPlatoro rule calls the function ReducedDiversionRequestsBelowPlatoro. This function compares the water right (total possible request) to the (modified) initial request, and the differences if any across all water users downstream of Platoro are summed to a single value.

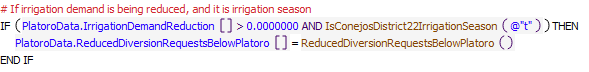
This rule fires if the ModifyConejosInitialRequests rule has fired successfully as checked with reference to the predefined HasRuleFiredSuccessfully function

Model slots written by rule:  
1. PlatoroData.ReducedDiversionRequestsBelowPlatoro

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
5/6/2015: Jesse Roach. Created rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**IsConejosDistrict22IrrigationSeason
* **RPL Object Icon**ReducedDiversionRequestsBelowPlatoro
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon2.4 RPL Object IconComputePlatoroCWCDDemand

Rule Purpose:  
The rule ComputePlatoroCWCDDemand calls the function PlatoroComputedCWCDDemand which in the absence of direct input data calculates the total CWCD (conservation storage) release demand from Platoro. This total conservation storage demand is calculated as the sum of the water rights based demand, additional release of conservation storage, less direct flow storage added, plus direct flow storage released. The rule ComputePlatoroCWCDDemand also records the release from direct flow storage for accounting purposes.

Rule Logic: Execution Constraint logic is at end of explanation.  
The rule calls the functions PlatoroReleaseFromDirectFlowStorage and PlatoroComputedCWCDDemand. The latter function in turn calls the functions PlatoroWaterRightsRelease, PlatoroCWCDRelease, MaxPlatoroDirectFlowStorage, and PlatoroReleaseFromDirectFlowStorage to sum the total conservation storage based release demand from Platoro in the current timestep.

This rule fires if has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND CWCD demand has not been computed

Model slots written by rule:  
1. PlatoroData.ComputedPlatoroCWCDDemand  
2. ComputedDeliveries.DirectFlowStorageReleaseFromPlatoro

List of key model objects with slots read by the rule or child functions:  
1. Platoro  
2. PlatoroData  
3. CompactCalculations

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/18/2018: Jesse Roach. Removed the CheckThisResPhysicalConstraint check as this function is used in the next rule, so this call is redundant.  
2/13/2018: Jesse Roach. Updated PlatoroComputedCWCDDemand function to use MaxPlatoroDirectFlowStorage() function instead of PlatoroDirectFlowStorageTarget()  
5/4/2015: Jesse Roach. Significant rework of PlatoroComputedCWCDDemand function to incorporate direct flow storage. Previous to this, function simply read in a schedule of input data.  
~2013: Craig Boroughs Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**PlatoroComputedCWCDDemand
* **RPL Object Icon**PlatoroReleaseFromDirectFlowStorage
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon2.5 RPL Object IconSetPlatoroCWCDDemand

Rule Purpose:  
The rule SetPlatoroCWCDDemand sets the Conejos Water Conservation District (CWCD) demand to the computed demand constrained to be between the minimum and maximum release that is hydraulically possible given the storage and the hydraulic properties of the outlet works.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule simply constrains the CWCD demand computed in the previous rule to a value that can be passed through the outlet works of Platoro.

This rule fires if has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND CWCD demand has not been defined

Model slots written by rule:  
1. PlatoroData.CWCDDemand

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
~2013: Craig Boroughs Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**CheckThisResPhysicalConstraints
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon2.6 RPL Object IconSetPlatoroOutflow

Rule Purpose:  
The rule SetPlatoroOutflow sets Platoro outflow to specified input if it exists and otherwise sets it to the Conejos Water Conservancy District release demand calculated (and limited by Platoro physical constraints) previously.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule uses an if statement to set Platoro.Outflow in the current timestep to a user input value if it exists (in the slot PlatoroData.TotalOutflow), or otherwise to the previously defined conservation storage release (in the slot PlatoroData.CWCDDemand).

This rule fires if has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND Platoro.Outflow has not been defined

Model slots written by rule:  
1. Platoro.Outflow

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
6/20/19: Marc Sidlow. To be consistent with other reservoirs, changed the PlatoroData.TotalOutflow slot to PlatoroData.TotalOutflowDirectlyInputToOverrideRules slot and use the TotalOutflow (Platoro) function  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Removed CheckThisResPhysicalConstraints function call for PlatoroData.CWCDDemand as that demand has already been constrained by that function.  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

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

### Control Display Icon2.7 RPL Object IconReleaseRelinquishedWaterByYearEnd

Rule Purpose:  
This rule releases water stored under Article VII in Platoro during a given irrigation season if Colorado doesn't have sufficient relinquished credits to carry the stored water to the next calendar year. This rule doesn't fire until after irrigation season, and is set up to release any water stored under Article VII for which credits are not available between the end of irrigation season and the end of the calendar year.

Rule Logic: Execution Constraint logic is at end of explanation.  
Platoro.Outflow is increased if the previously calculated value in Platoro.Outflow (from either direct user input or conservation storage release rules) is not sufficient to move the water stored under Article VII conditions (Platoro^RelinquishedWater.Storage) in excess of available relinquished credits (RelinquishedCreditsEmergencyDroughtWater.RelinquishedCOCredits). The total water to be moved is spread out over the remainder of the year by dividing the total to be moved by the remaining time in the year.

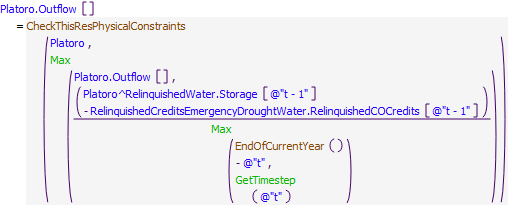
This rule fires if it has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND if the timestep is at or after the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND if the timestep is after June 30 and after the end of of the irrigation season in Colorado according to the user defined function IsIrrigationSeasonCO, which is currently from April 1 to October 31 AND if the water stored under Article VII conditions (Platoro^RelinquishedWater.Storage) is greater than the available relinquished credits (RelinquishedCreditsEmergencyDroughtWater.RelinquishedCOCredits) minus the Compact Accounting Threshold (which is currently 50 acre-ft)

Model slots written by rule:  
1. Platoro.Outflow

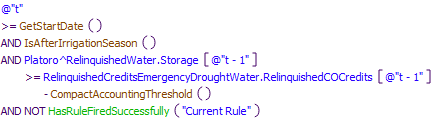
List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Platoro  
3. RelinquishedCreditsEmergencyDroughtWater

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Moved conditionals from IF statement into execution constraints and removed the IF statement.  
5/7/2015: Jesse Roach. Created rule.

Statements

****

Execution Constraint

****

Referenced Functions

* **RPL Object Icon**CheckThisResPhysicalConstraints
* **RPL Object Icon**EndOfCurrentYear
* **RPL Object Icon**IsAfterIrrigationSeason
* **RPL Object Icon**CompactAccountingThreshold
* **RPL Object Icon**Max
* **RPL Object Icon**GetTimestep
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon2.8 RPL Object IconComputePlatoroOutflowsForMinFlowAndMinPool

Rule Purpose:  
This rule calculates the minimum required outflow from Platoro for instream flow requirements, and also calculates the maximum outflow from Platoro to stay above the minimum pool level. This rule does not set a new value on the Platoro.Outflow slot. The calculated values are set on the PlatoroData.ComputedPlatoroMinOutflow and PlatoroData.ComputedPlatoroToStayAboveMinimumPool slots.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule fires on all timesteps after the date specified in the ModelRunTypeTriggers.RulebasedSimulationStartDay by calling two different functions specific to calculating Platoro minimum release requirements and Platoro maximum possible release to not go below minimum pool requirements. This rule does not set a new value on the Platoro.Outflow slot. The calculated values are set on the PlatoroData.ComputedPlatoroMinOutflow and PlatoroData.ComputedPlatoroToStayAboveMinimumPool slots for reference by a subsequent rule.

This rule fires if the current timestep is greater than or equal to the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function and If the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. PlatoroData.ComputedPlatoroMinOutflow  
2. PlatoroData.ComputedPlatoroToStayAboveMinimumPool

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/1/2018: Jesse Roach. Simplified and combined two rules to calculate minimum release and maximum release to maintain minimum pool elevation rules and renamed. Old rule names: ComputePlatoroMinOutflow and ComputePlatoroOutflowToStayAboveMinimumPool.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**PlatoroComputedMinOutflow
* **RPL Object Icon**OutflowToStayAbovePlatoroMinimumStorage
* **RPL Object Icon**Max
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon2.9 RPL Object IconAdjustPlatoroOutflow

Rule Purpose:  
This rule adjusts Platoro outflow if either the previously calculated conservation release-based outflow is less than the minimum release requirements, or the previously calculated release would result in the pool going below the minimum pool level. The adjusted flow is constrained to physically possible releases from Platoro. If the minimum release would result in a pool elevation below the minimum required, the minimum release is reduced to maintain the higher priority pool elevation constraint.

Rule Logic: Execution Constraint logic is at end of explanation.  
The rule only fires if the calculated Platoro outflow is below the minimum required (too small a release), or would take the reservoir below the minimum pool (too large a release). The larger of Platoro.Outflow and PlatoroData.ComputedPlatoroMinOutflow satisfies the minimum flow requirements, while the smaller of that value and the PlatoroData.ComputedPlatoroToStayAboveMinimumPool value satisfies the minimum pool storage requirement. The pool requirement has higher priority than the release requirement.

This rule fires if it has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND if the timestep is equal to or past the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND either the Outflow from Platoro is greater than outflow necessary to stay above the minimal pool elevation OR the outflow is less than the minimum outflow

Model slots written by rule:  
1. Platoro.Outflow

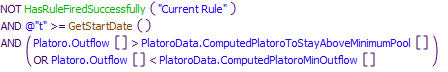
List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. PlatoroData  
3. Platoro

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/1/2018: Jesse Roach. Combined minimum release rule (formerly called SetPlatoroMinOutflow) and maximum release to maintain minimum pool elevation rule (called SetPlatoroOutflowToStayAboveMinimumPool) and renamed.

Statements

****

Execution Constraint

****

Referenced Functions

* **RPL Object Icon**CheckThisResPhysicalConstraints
* **RPL Object Icon**Min
* **RPL Object Icon**Max
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

## Control Display Icon3 RPL Object IconPlatoro Flood Control Policy

This policy group sets Platoro releases to implement Platoro flood control policy. Flood control operations at Platoro are determined throughout the months of March through July of each year based on the USACE January 1963 Flood-Control Storage Reservation Diagram (Diagram) and a spreadsheet called FLODSP7A.xlsx was developed to help automate the use of the Diagram. The flood control diagram specifies an evacuation period from March 1 through May 9, and a filling period from May 10 through July 31. The evacuation period uses forecasted and historical runoff volumes to reach a specified storage on May 10, after which time Platoro can fill depending on the remaining forecasted runoff through July. URGWOM flood control operations at Platoro as implemented in this Policy Group follow this methodology.

Policy Group Change Log (newest changes at the top):   
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Added Group Description and Group Notes fields.  
~2013: Craig Boroughs Created this Policy Group and the rules within it.

### Control Display Icon3.1 RPL Object IconComputePlatoroChangeInStorage

Rule Purpose:  
This rule calculates the change in Storage in Platoro from the previous timestep to the current timestep as of the time it is called, and executes from March 1st to December 31st of each year. The slot set by this rule is called by the "ComputeCumulativeMogoteNaturalVolume" rule to calculate the total natural flow at the Mogote gage.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the change in storage slot in the Platoro Flood Diagram data object by comparing Platoro.Storage[@"t"] to Platoro.Storage[@"t-1"].  
The rule only fires on or after March 1.

Model slots written by rule:  
1. Platoro Flood Diagram.PlatoroChangeInStorage

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
~2013: Craig Boroughs Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**ResStorageChange

### Control Display Icon3.2 RPL Object IconComputePlatoroPreviousChangeInStorage

Rule Purpose:  
This rule calculates the change in Storage in Platoro from the previous timestep to the current timestep as of the time it is called, and executes from March 1st to December 31st of each year. The slot set by this rule is called by the "ComputeCumulativeMogoteNaturalVolume" rule to calculate the total natural flow at the Mogote gage.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the change in storage slot in the Platoro Flood Diagram data object by comparing Platoro.Storage[@"t"] to Platoro.Storage[@"t-1"].  
The rule only fires on or after March 1.

Model slots written by rule:  
1. Platoro Flood Diagram.PlatoroChangeInStorage

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
~2013: Craig Boroughs Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**PlatoroPreviousResStorageChange

### Control Display Icon3.3 RPL Object IconComputeMogoteForecastVolume

Rule Purpose:  
This rule calculates the Forecast Volume, if directly input for an AOP model run, it uses the input forecast volume for Mogote, if not, it estimates the total flow that would have gone past Mogote in the absence of Platoro or any diversions (sum of inflows corrected for storage change at Platoro) starting in March through the current timestep for use in flood control calculations.

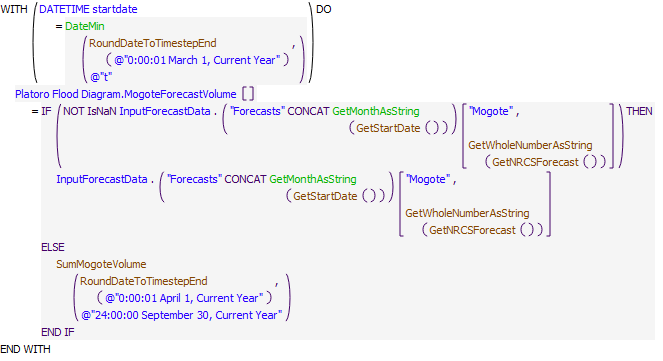
Rule Logic: Execution Constraint logic is at end of explanation.  
The rule checks if there is an input value for the Mogote forecast location for the month being forecasted, if so set the value of the input forecast amount, if not, sum the Platoro inflows and Local Inflows between Platoro and Mogote, using the user defined function - SumMogoteVolume.  
The rule only fires on or after March 1 and on or before September 30 each year.

Model slots written by rule:  
1. Platoro Flood Diagram.ComputeMogoteFForecastVolume

List of key model objects with slots read by the rule or child functions:  
1. Platoro  
2. LocalInflowsAtMogote  
3. Platoro Flood Diagram

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
09/26/22: Marc Sidlow. Created rule and added description field and notes field.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**RoundDateToTimestepEnd
* **RPL Object Icon**SumMogoteVolume
* **RPL Object Icon**DateMin
* **RPL Object Icon**GetMonthAsString
* **RPL Object Icon**GetWholeNumberAsString
* **RPL Object Icon**GetStartDate
* **RPL Object Icon**GetNRCSForecast

### Control Display Icon3.4 RPL Object IconComputeMogoteVolume

Rule Purpose:  
This rule calculates the flow that passes Mogote and converts to volume in the Platoro Flood Diagram data object, and will be uesd in subsequent Platoro rules.

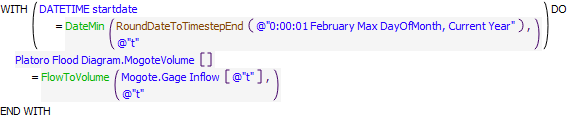
Rule Logic: Execution Constraint logic is at end of explanation.  
The rule converts the Mogote.Gage Inflow from flow in cfs to volume in ac-ft using the pre-defined function FlowToVolume.  
The rule only fires on or after March 1.

Model slots written by rule:  
1. Platoro Flood Diagram.MogoteVolume

List of key model objects with slots read by the rule or child functions:  
1. Mogote  
2. Platoro Flood Diagram

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
09/26/22: Marc Sidlow. Created rule and added description field and notes field.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**RoundDateToTimestepEnd
* **RPL Object Icon**DateMin
* **RPL Object Icon**FlowToVolume

### Control Display Icon3.5 RPL Object IconComputePreviousMogoteVolume

Rule Purpose:  
This rule calculates the flow that passes Mogote and converts to volume in the Platoro Flood Diagram data object, and will be uesd in subsequent Platoro rules.

Rule Logic: Execution Constraint logic is at end of explanation.  
The rule converts the Mogote.Gage Inflow from flow in cfs to volume in ac-ft using the pre-defined function FlowToVolume.  
The rule only fires on or after March 1.

Model slots written by rule:  
1. Platoro Flood Diagram.MogoteVolume

List of key model objects with slots read by the rule or child functions:  
1. Mogote  
2. Platoro Flood Diagram

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
09/26/22: Marc Sidlow. Created rule and added description field and notes field.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**RoundDateToTimestepEnd
* **RPL Object Icon**DateMin
* **RPL Object Icon**FlowToVolume

### Control Display Icon3.6 RPL Object IconComputeNaturalMogoteVolume

Rule Purpose:  
This rule calculates the dailyl flow volume that passes Mogote inflows corrected for storage change at Platoro starting in March through the current timestep for use in flood control calculations.

Rule Logic: Execution Constraint logic is at end of explanation.  
The rule computes at the current timestep and sums Mogote volume and storage change at Platoro.  
The rule only fires on or after March 1.

Model slots written by rule:  
1. Platoro Flood Diagram.ComputeMogoteNaturalVolume

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
09/26/22: Marc Sidlow. Created new rule for comparing slot series to column in Platoro Flood Operations spreadsheet.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**RoundDateToTimestepEnd
* **RPL Object Icon**DateMin

### Control Display Icon3.7 RPL Object IconComputeCumulativeMogoteNaturalVolume

Rule Purpose:  
This rule calculates the total flow that would have gone past Mogote in the absence of Platoro or any diversions (sum of inflows corrected for storage change at Platoro) starting in March through the current timestep for use in flood control calculations.

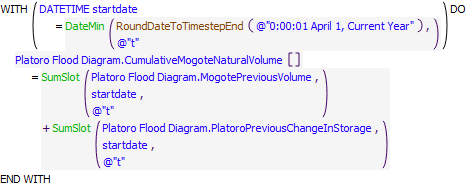
Rule Logic: Execution Constraint logic is at end of explanation.  
The rule loops through all timesteps in March through the current timestep and sums Platoro inflows and Local Inflows between Platoro and Mogote and storage change at Platoro.  
The rule only fires on or after March 1.

Model slots written by rule:  
1. Platoro Flood Diagram.CumulativeMogoteNaturalVolume

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
09/26/22: Marc Sidlow. Now using the previuosly computed Mogote Volume and correcting for change in storage at Platoro, and now computing beginning April 1.  
2/13/19: Jesse Roach. Added description field and updated notes field.  
11/17/17: Jesse Roach. Generalized for daily or monthly timestep  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**RoundDateToTimestepEnd
* **RPL Object Icon**SumSlot
* **RPL Object Icon**DateMin

### Control Display Icon3.8 RPL Object IconComputedPlatoroOutflowToGetBelowFloodPool

Rule Purpose:  
This rule calculates the release from Platoro in the current timestep that would be necessary to empty the flood pool.

Rule Logic: Execution Constraint logic is at end of explanation.  
The rule calls a function which uses the RiverWare function SolveOutflow to calculate the outflow in the current timestep necessary to get the pool elevation 0.01 feet below the BottomOfFlood elevation stored in the PlatoroData Data Object.

This rule fires if the current timestep is greater than or equal to the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function and If the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function..

Model slots written by rule:  
1. PlatoroData.ComputedPlatoroToGetBelowFloodPool

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Removed CheckThisResPhysicalConstraints because that function is called before this slot set to outflow.  
~2013: Craig Boroughs Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

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

### Control Display Icon3.9 RPL Object IconSetPlatoroOutflowToGetBelowFloodPool

Rule Purpose:  
This rule releases any water in the flood pool within the physical constraints of the reservoir.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule increases the Platoro.Outflow slot to the value calculated by the "ComputePlatoroOutfllowToGetBelowFloodPool" rule if the pool elevation is above the bottom of the flood pool elevation.

This rule fires if the current timestep is greater than or equal to the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function and If the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND , if Platoro is storing water in the flood pool, AND if the current Platoro outflow is less than the amount necessary to evacuate the flood pool.

Model slots written by rule:  
1. Platoro.Outflow

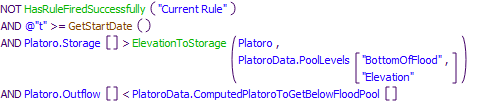
List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Platoro  
3. Platoro.Data

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Moved storage below flood pool conditional to execution constraints and added a max function to execution constraints to make sure the rule doesn't reduce Platoro.Outflow.  
~2013: Craig Boroughs Created the rule.

Statements

**Statements**

Execution Constraint

****

Referenced Functions

* **RPL Object Icon**CheckThisResPhysicalConstraints
* **RPL Object Icon**ElevationToStorage
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon3.10 RPL Object IconComputePlatoroForecastIndex

Rule Purpose:  
This rule computes the Platoro Forecast Index as a function of the current date and estimated naturalized volume past Mogote from April through July. The forecast index varies between 0.7 and 3.6 and when multiplied by 100,000 AF approximates the total naturalized volume that would flow past Mogote from April through July.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule computes the Platoro Forecast Index by calling two functions. The first function, PlatoroForecastIndex() returns a volume of water associated with the estimated volume of naturalized flow past Mogote from April through July. This volume is used as an argument to the function ComputeIndex, which translates the volume to an index value between 0.7 and 3.6.

This rule fires if the current timestep is greater than or equal to the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function and If the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND the Platoro Flood Diagram.Index model slot value is unassigned..

Model slots written by rule:  
1. Platoro Flood Diagram.Index

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Platoro Flood Diagram  
3. Platoro  
4. LocalInflowsAtMogote

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
09/26/22: Marc Sidlow. Modified Execute Rule Only When logic to start rule on March 1 vs. GetStartDate user function.  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Rule moved down 1 place in priority so that the slot it writes (Platoro Flood Diagram.Index) can be used by the rule ComputeAllowableStorageReservation.  
~2013: Craig Boroughs Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**PlatoroForecastIndex
* **RPL Object Icon**ComputeIndex
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon3.11 RPL Object IconComputeAllowableStorageReservation

Rule Purpose:  
This rule computes the maximum storage allowed in Platoro as a function of date and projected inflows to the Conejos above Mogote. A flood control diagram from the Platoro Water Control Manual is used to determine this value.

Rule Logic: Execution Constraint logic is at end of explanation.This rule calculates the allowable storage by calling the function MaximumAllowableStorage . This function performs a table lookup into a table that lists maximum allowable storage at Platoro as a function of date and the PlatoroForecastIndex.

This rule fires at or after the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND if the timestep on or between March 1 and July 31 AND if it has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND the value of the model slot PlatoroFloodDiagram.AllowableStorageReservation is unassigned.

Model slots written by rule:  
1. Platoro Flood Diagram.AllowableStorageReservation

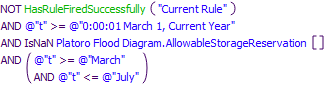
List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Platoro Flood Diagram

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
09/26/22: Marc Sidlow. Modified Execute Rule Only When logic to start rule on March 1 vs. GetStartDate user function.  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Moved the Boolean expression out of an IF statement and into the execution constraints and got rid of the IF statement.  
~2013: Craig Boroughs Created the rule.

Statements

**Statements**

Execution Constraint

****

Referenced Functions

* **RPL Object Icon**MaximumAllowableStorage
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon3.12 RPL Object IconComputeStorageToVoidOrFill

Rule Purpose:  
This rule computes the maximum storage allowed in Platoro as a function of date and projected inflows to the Conejos above Mogote. A flood control diagram from the Platoro Water Control Manual is used to determine this value.

Rule Logic: Execution Constraint logic is at end of explanation.This rule calculates the allowable storage by calling the function MaximumAllowableStorage . This function performs a table lookup into a table that lists maximum allowable storage at Platoro as a function of date and the PlatoroForecastIndex.

This rule fires at or after the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND if the timestep on or between March 1 and July 31 AND if it has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND the value of the model slot PlatoroFloodDiagram.AllowableStorageReservation is unassigned.

Model slots written by rule:  
1. Platoro Flood Diagram.AllowableStorageReservation

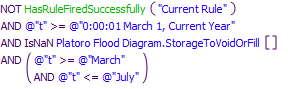
List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Platoro Flood Diagram

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
09/26/22: Marc Sidlow. Modified Execute Rule Only When logic to start rule on March 1 vs. GetStartDate user function.  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Moved the Boolean expression out of an IF statement and into the execution constraints and got rid of the IF statement.  
~2013: Craig Boroughs Created the rule.

Statements

**Statements**

Execution Constraint

****

Referenced Functions

* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon3.13 RPL Object IconComputeProratedDailyReleaseVolumeToReachTarget

Rule Purpose:  
This rule computes the maximum storage allowed in Platoro as a function of date and projected inflows to the Conejos above Mogote. A flood control diagram from the Platoro Water Control Manual is used to determine this value.

Rule Logic: Execution Constraint logic is at end of explanation.This rule calculates the allowable storage by calling the function MaximumAllowableStorage . This function performs a table lookup into a table that lists maximum allowable storage at Platoro as a function of date and the PlatoroForecastIndex.

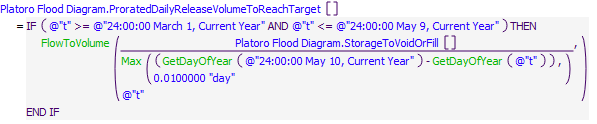
This rule fires at or after the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND if the timestep on or between March 1 and July 31 AND if it has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND the value of the model slot PlatoroFloodDiagram.AllowableStorageReservation is unassigned.

Model slots written by rule:  
1. Platoro Flood Diagram.AllowableStorageReservation

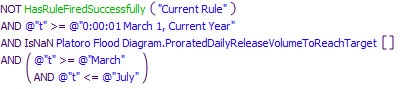
List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Platoro Flood Diagram

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
09/26/22: Marc Sidlow. Modified Execute Rule Only When logic to start rule on March 1 vs. GetStartDate user function.  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Moved the Boolean expression out of an IF statement and into the execution constraints and got rid of the IF statement.  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

****

Referenced Functions

* **RPL Object Icon**Max
* **RPL Object Icon**GetDayOfYear
* **RPL Object Icon**FlowToVolume
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon3.14 RPL Object IconComputePlatoroOutflowToGetBelowAllowableStorageReservation

Rule Purpose:  
This rule computes the release necessary to get Platoro storage below the maximum storage allowed during flood control operations (March through July).

Rule Logic: Execution Constraint logic is at end of explanation.  
The rule calls the function FlowToGetBelowAllowableStorageReservation and limits the result of that function if necessary based on the physical constraints of the reservoir outlet works.

This rule fires at or after the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND if it has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND if the value of the model slot Platoro Flood Diagram.AllowableStorageReservation is unassigned.

Model slots written by rule:  
1. Platoro Flood Diagram.ComputedPlatoroOutflowToGetBelowAllowableStorage

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Removed the CheckThisResPhysicalConstraints function call as that call is used again when the slot set by this rule is used to set actual outflows.  
~2013: Craig Boroughs Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

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

### Control Display Icon3.15 RPL Object IconSetPlatoroOutflowToGetBelowAllowableStorageReservation

Rule Purpose:  
This rule increases the slot Platoro.Outflow if necessary to bring Platoro storage below the maximum allowable storage during flood control operations (March through July).

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule uses an IF statement to check if the flow necessary to get below the allowable storage is greater than the current outflow, and if it is, sets Platoro.Outflow to the larger value within the physical constraints of Platoro's outlet works.

This rule fires if the current timestep is greater than or equal to the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND if the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND when there is a maximum allowable storage value AND the computed outflow to get below the allowable storage is greater than the current outflow

Model slots written by rule:  
1. Platoro.Outflow

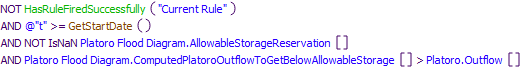
List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Platoro Flood Diagram  
3. Platoro

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Replaced the call to the FlowToGetBelowAllowableStorageReservation() function with Platoro Flood Diagram.ComputedPlatoroOutflowToGetBelowAllowableStorage, the slot set by that same function previously in the timestep, and pulled that changed Boolean comparison out of the rule body and into the Execution Constraints, and deleted IF statement in the rule body.  
~2013: Craig Boroughs Created the rule.

Statements

**Statements**

Execution Constraint

****

Referenced Functions

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

### Control Display Icon3.16 RPL Object IconRecordMaxOutflowsForChannelCapacities

Rule Purpose:  
This rule calculates the maximum release from Platoro possible without contributing to flow greater than channel capacity at Mogote or La Sauces. Minimum Platoro release constraints are maintained regardless of downstream conditions.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule calculates the maximum Platoro outflow for La Sauces and Mogote separately by calling separate functions for each location. Each value is written to a separate slot.

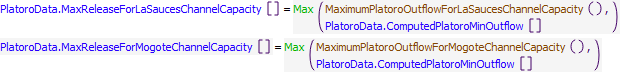
This rule fires if the current timestep is greater than or equal to the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function and If the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. PlatoroData.MaxReleaseForMogoteChannelCapacity  
2. PlatoroData.MaxReleaseForLaSaucesChannelCapacity

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. PlatoroData  
3. LocalInflowsAtMogote  
4. LocalInflowsConejosAtLaSauces  
5. MogoteToLaSaucesLossesAndLag  
6. CompactCalculations  
7. RioLosPinosAtOrtiz  
8. RioSanAntonioAtOrtiz  
9. SanAntonioRiverLossesAndLag  
10. LocalInflowsToSanAntonioRiverAtManassa  
11. PlatoroToMogoteLossesAndLag

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Got rid of CheckThisResPhysicalConstraints function calls as that call is used if the slots set by this rule are used to set actual outflows. Also substituted calls to PlatoroData.ComputedPlatoroMinOutflow[] for function calls to the PlatoroComputedMinOutflow function.  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**MaximumPlatoroOutflowForMogoteChannelCapacity
* **RPL Object Icon**MaximumPlatoroOutflowForLaSaucesChannelCapacity
* **RPL Object Icon**Max
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon3.17 RPL Object IconPlatoroOutflowRestrictions

Rule Purpose:  
This rule reduces Platoro outflow if necessary based on downstream channel capacity constraints.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule reduces the outflow from Platoro if necessary by selecting the more constraining channel capacity release with the PlatoroRelaseForChannelCapacity function. The rule constrains the outflow based on the physical constraints of the Platoro outlet works.

This rule fires if the current timestep is greater than or equal to the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function and If the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. Platoro.Outflow

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
~2013: Craig Boroughs Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**CheckThisResPhysicalConstraints
* **RPL Object Icon**PlatoroReleaseForChannelCapacity
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**GetStartDate

### Control Display Icon3.18 RPL Object IconPlatoroFloodControl

Rule Purpose:  
This rule attempts to avoid an unregulated spill by increasing the release if possible without violating channel capacity constraints.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule checks for an unregulated spill, and if there is an unregulated spill, calls the function PlatoroFloodControlRelease to potentially increase the release.

This rule fires when there is an unregulated spill from Platoro on or after the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function.

Model slots written by rule:  
1. Platoro.Outflow

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Moved Platoro.Unregulated Spill[]>0 comparison into execution constraints and got rid of IF function in rule body.  
~2013: Craig Boroughs Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**CheckThisResPhysicalConstraints
* **RPL Object Icon**PlatoroFloodControlRelease
* **RPL Object Icon**GetStartDate

## Control Display Icon4 RPL Object IconPlatoro Storage Accounting

This policy group sets Platoro accounts. Accounts are set after the physical reservoir has solved. There are three types of water stored in Platoro: Direct Flow Storage, Relinquished Water, and Conejos Project Storage. There is a fourth storage account in Platoro which is Conejos, which is used as a pass through, or the source of transfers to the other three accounts. This Policy Group is used to determine what portion of water stored and released from Platoro should be assigned to each of the accounts.

Policy Group Change Log (newest changes at the top):  
Date: Who. What   
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Added Group Description and Group Notes fields.  
April/2015: Jesse Roach. Rearranged rules so that accounting on the reservoir takes place after the physical reservoir solves. Added the rule UpdateCOCreditStorage to the Policy Group.  
~2013: Craig Boroughs Created the Policy Group

### Control Display Icon4.1 RPL Object IconSetTransferToRelinquishedWaterStorage

Rule Purpose:  
This rule tracks any water added to storage at Platoro that was not added as Direct Flow Storage. Such water is tagged as Relinquished Water stored in Platoro, and must be evacuated before the end of the calendar year, or offset with relinquishment credits to keep in storage into the next calendar year. This is based on Colorado's interpretation of Article VII as an annual accounting constraint, as opposed to New Mexico's interpretation of Article VII as a real time requirement.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the slot ConejosPlatoroToRelinquishedWaterPlatoro.Supply by calling the function CORelinquishedInflow. This function transfers any water added to storage under ArticleVII conditions into the Relinquished Water pool at Platoro, unless that water was stored as Direct Flow Storage water.

This rule fires if the timestep is at or after the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND if ConejosPlatoroToRelinquishedWaterPlatoro.Supply model slot value is unassigned.

Model slots written by rule:  
1. ConejosPlatoroToRelinquishedWaterPlatoro.Supply

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Platoro  
3. PlatoroData  
4. RioGrandeCompact

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
5/7/2015: Jesse Roach. CORelinquishedInflow function simplified to not track available space since Colorado only limits storage of Relinquished Water to available credits on the last day of the year.  
4/29/2015: Jesse Roach. Rule priority adjusted to execute after physical reservoir solves.  
~2013: Craig Boroughs Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**CORelinquishedInflow
* **RPL Object Icon**GetStartDate

### Control Display Icon4.2 RPL Object IconComputeCORelinquishedWaterRelease

Rule Purpose:  
This rule computes the amount of water released from Platoro in the current timestep that was water initially stored under Article VII restrictions (Relinquished Water). There are three types of water stored in Platoro: Direct Flow Storage, Relinquished Water, and Conejos Project Storage. There is a fourth storage account in Platoro which is Conejos, but this account is used as a pass through, or the source of transfers to the other three accounts. Downstream demands for stored water are served first by Direct Flow Storage (during a specific period of the year), and next by Relinquished Water.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the slot ComputedDeliveries.RelinquishmentWaterReleaseFromPlatoro by calling the function CORelinquishedOutflow. This function calculates the portion of releases from storage in the current timestep that can be assigned to the Relinquished Water pool.

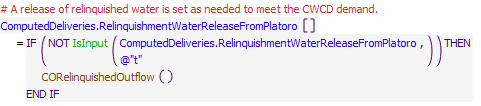
This rule fires if the timestep is at or after the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND if ComputedDeliveries.RelinquishmentWaterReleaseFromPlatoro model slot value is unassigned.

Model slots written by rule:  
1. ComputedDeliveries.RelinquishmentWaterReleaseFromPlatoro

List of key model objects with slots read by the rule or child functions:  
1. ComputedDeliveries  
2. ModelRunTypeTriggers  
3. Platoro  
4. PlatoroData  
5. RioGrandeCompact

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
4/29/2015: Jesse Roach. Rule priority adjusted to execute after physical reservoir solves.  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**CORelinquishedOutflow
* **RPL Object Icon**IsInput
* **RPL Object Icon**GetStartDate

### Control Display Icon4.3 RPL Object IconSetTransferToConejosProjectStorage

Rule Purpose:  
This rule sets water stored in Platoro outside of Article VII restrictions, and not including water stored in lieu of diversion in the same timestep (Direct Flow Storage water) as Conejos Project water. The amount of Conejos Project water that can be stored is limited to a maximum account storage specified in Platoro for Conejos Project Storage.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the slot ConejosPlatoroToConejosProjectStoragePlatoro.Supply by calling the function ConejosProjectInflow. This function computes the transfer of Conejos inflows to Conejos Project storage. Positive storage changes are stored as Conejos Project water when  
Article VII is NOT in effect, up to an account limit stored in PlatoroData.MaxAccountStorages.

This rule fires if the timestep is at or after the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND if ConejosPlatoroToConejosProjectStoragePlatoro.Supply model slot value is unassigned.

Model slots written by rule:  
1. ConejosPlatoroToConejosProjectStoragePlatoro.Supply

List of key model objects with slots read by the rule or child functions:  
1. ModelRunTypeTriggers  
2. Platoro  
3. PlatoroData  
4. RioGrandeCompact

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
4/29/2015: Jesse Roach. Rule priority adjusted to execute after physical reservoir solves.  
~2013: Craig Boroughs Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**ConejosProjectInflow
* **RPL Object Icon**GetStartDate

### Control Display Icon4.4 RPL Object IconComputeConejosProjectStorageRelease

Rule Purpose:  
This rule determines what portion if any of Platoro's release in a given timestep should be from the Conejos Project water account. There are three types of water stored in Platoro: Direct Flow Storage, Relinquished Water, and Conejos Project Storage. There is a fourth storage account in Platoro which is Conejos, but this account is used as a pass through, or the source of transfers to the other three accounts. Downstream demands in excess of by-passed inflows are served first by Direct Flow Storage (during a specific period of the year), next by Relinquished Water, and finally by Conejos Project Storage.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the slot ComputedDeliveries.ConejosProjectReleaseFromPlatoro by calling the function ConejosProjectStorageOutflow. This function calculates the portion of releases from storage in the current timestep that should come from the Conejos Project Storage pool.

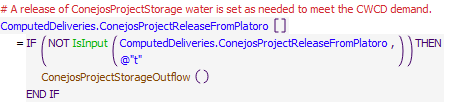
This rule fires if the timestep is at or after the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND if ComputedDeliveries.ConejosProjectReleaseFromPlatoro model slot value is unassigned.

Model slots written by rule:  
1. ComputedDeliveries.ConejosProjectReleaseFromPlatoro

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
4/29/2015: Jesse Roach. Rule priority adjusted to execute after physical reservoir solves.  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**ConejosProjectStorageOutflow
* **RPL Object Icon**IsInput
* **RPL Object Icon**GetStartDate

### Control Display Icon4.5 RPL Object IconSetAvailableCORelinquishedCompactCredits

Rule Purpose:  
This rule adjusts Colorado Relinquished Compact Credits twice per year, once when downstream credits in Elephant Butte above a given threshold are transferred to Colorado for the right to carry-over storage in Platoro during Article VII (Colorado interpretation of Article VII), and a second time at the end of the calendar year when the relinquished credits are reduced by an amount equal to Relinquished Water (water stored under Article VII restrictions) that is in storage in Platoro. The end of year reduction occurs with a transfer of water from the Relinquished Water account to the Conejos Project Storage account, which is accomplished with the corollary rule SetTransferFromRelinquishedWaterStorage.

Rule Logic: Execution Constraint logic is at end of explanation.  
This takes the previous value in the slot RelinquishedCreditsEmergencyDroughtWater.RelinquishedCOCredits and if relinquishments are being modeled, and it is the "Relinquishment Date", adds Colorado relinquished credit if any. On the last timestep of the year, this rule takes the previous value in the slot and removes a volume equivalent to the amount of water in the Relinquished Water account at the end of the year.

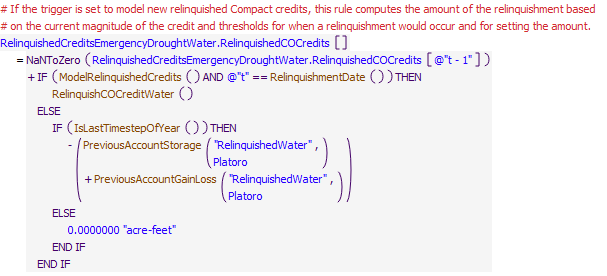
This rule fires if the timestep is at or after the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND if RelinquishedCreditsEmergencyDroughtWater.COCreditStorage model slot value is unassigned.

Model slots written by rule:  
1. RelinquishedCreditsEmergencyDroughtWater.RelinquishedCOCredits

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**IsLastTimestepOfYear
* **RPL Object Icon**RelinquishmentDate
* **RPL Object Icon**RelinquishCOCreditWater
* **RPL Object Icon**ModelRelinquishedCredits
* **RPL Object Icon**PreviousAccountGainLoss
* **RPL Object Icon**GetStartDate
* **RPL Object Icon**PreviousAccountStorage

### Control Display Icon4.6 RPL Object IconSetTransferFromRelinquishedWaterStorage

Rule Purpose:  
This rule moves water that was stored under Article VII restrictions in a given year into the Conejos Project Storage pool to start the next year. This transfer requires a reduction by the same amount of Colorado's relinquished credits which is accomplished with the corollary rule SetAvailableCORelinquishedCompactCredits.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the slot RelinquishedWaterPlatoroToConejosProjectStoragePlatoro.Supply to the amount of Relinquished Water in storage at Platoro at the end of the calendar year. This value is estimated by taking the amount of Relinquished Water at the end of the previous timestep and adding the loss and gains to that account in the previous timestep.

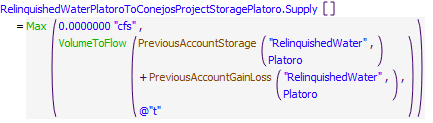
This rule fires on the last timestep of the year if it is greater than or equal to the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND If the rule has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. RelinquishedWaterPlatoroToConejosProjectStoragePlatoro.Supply

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Moved IsLastTimestepOfYear function into Execution Constraints and got rid of IF statement in body of the rule.  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**IsLastTimestepOfYear
* **RPL Object Icon**Max
* **RPL Object Icon**VolumeToFlow
* **RPL Object Icon**HasRuleFiredSuccessfully
* **RPL Object Icon**PreviousAccountGainLoss
* **RPL Object Icon**GetStartDate
* **RPL Object Icon**PreviousAccountStorage

### Control Display Icon4.7 RPL Object IconSetTransferToDirectFlowStorage

Rule Purpose:  
This rule transfers water from the Conejos account into storage in Platoro in the Platoro Direct Flow Storage account. This storage may occur during specific periods of the year when water rights holders downstream choose to store water in Platoro that they would otherwise be in priority to divert downstream.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the slot ConejosPlatoroToDirectFlowStoragePlatoro.Supply by calling the function PlatoroDirectFlowStorageTarget. This function takes the smaller of water added to storage in Platoro, and the reduced diversion requests below Platoro (representing water rights holders choosing to store rather than divert). This rule sets the slot ConejosPlatoroToDirectFlowStoragePlatoro.Supply.

This rule fires if the timestep is at or after the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND if the ConejosPlatoroToDirectFlowStoragePlatoro.Supply model slot value is unassigned.

Model slots written by rule:  
1. ConejosPlatoroToDirectFlowStoragePlatoro.Supply

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
2/13/2018: Jesse Roach. Updated the function PlatoroDirectFlowStorageTarget due to accounting errors with direct flow storage account.  
~2015: Jesse Roach. Created the rule.

Statements

**Statements**

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**PlatoroDirectFlowStorageTarget
* **RPL Object Icon**GetStartDate

### Control Display Icon4.8 RPL Object IconUpdateCOCreditStorage

Rule Purpose:  
This rule reduces Colorado Credit in Elephant Butte when that credit water is exchanged for relinquished credits to be used to carry water stored in Platoro under Article VII restrictions into the next calendar year (Colorado interpretation of Article VII restrictions).

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets the value of the slot RelinquishedCreditsEmergencyDroughtWater.COCreditStorage in the current timestep as the value of that same slot in the previous timestep less any relinquishment of credits that occurs in the same timestep as stored in the slot RelinquishedCreditsEmergencyDroughtWater.RelinquishedCOCredits.  
This rule likely needs to be expanded to update COCreditStorage in Elephant Butte when Colorado overdelivers.

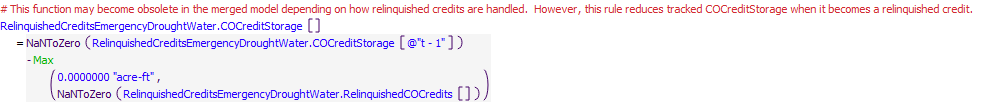
This rule fires if the timestep is at or after the beginning timestep for rulebased simulation as input by the model user to the RulebasedSimulationStartDate scalar slot in the ModelRunTypeTriggers data object and as identified with the user-defined GetStartDate function AND if RelinquishedCreditsEmergencyDroughtWater.COCreditStorage model slot value is unassigned.

Model slots written by rule:  
1. RelinquishedCreditsEmergencyDroughtWater.COCreditStorage

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Added Max function to ensure COCreditStorage can only be reduced by this rule.  
4/23/2015: Jesse Roach. Rule created because the series slot RelinquishedCreditsEmergencyDroughtWater.COCreditStorage which tracks Colorado Credit Storage in Elephant Butte, was not being updated when credit in EB was being exchanged for relinquished credits in Colorado.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**Max
* **RPL Object Icon**GetStartDate

## Control Display Icon5 RPL Object IconSet Colorado Diversions

This policy group does two main things. First, it sets water type on flows in the Colorado portion of the model. Water types for all inflows except those above Platoro are set based on the Compact Curtailment Percentages calculated for the Rio Grande and Conejos River systems in the CompactCalculations Data Object. Water types for releases from Platoro are set based on Platoro Storage Accounting results from the previous Policy Group. Second, this Policy Group sets the Colorado Diversion requests to zero during non-irrigation season, and based on water rights using the RiverWare Water Rights Solver functionality during irrigation season.

Policy Group Change Log (newest changes at the top):  
Date: Who. What   
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/13/2018: Jesse Roach. Added Group Description and Group Notes fields.  
8/8/2018: Jesse Roach. Created the Policy Group from four smaller policy groups. These four groups were most likely created in 2013 by Craig Boroughs in the original Colorado portion of URGWOM rules set. The four Policy Groups which were combined to form this Policy Group were  
1. SetCompactCurtailmentsVersusAllocatableFlow (3 rules)  
2. SetDiversionAccountsToZeroForNonIrrigationSeason (2 rules)  
3. SetDiversionAccountsWithWaterRightsSolver (2 rules)  
4. SetPhysicalDiversions (2 rules)  
~2013: Craig Boroughs Created the Policy Group(s)

### Control Display Icon5.1 RPL Object IconSetPlatoroPassthroughAccounts

Rule Purpose:  
This rule sets the supplies out of Platoro to the Compact Delivery and Allocatable Flow accounts based on Platoro Accounting. Water released from Platoro that is not included in specific Platoro Accounting is split into Compact Delivery and Allocatable Flow accounts based on the Conejos Compact Curtailment percentage. The Conejos Compact Curtailment percentage is calculated based on among other things, the cumulative departure of the delivery to the Colorado-New Mexico stateline from the Compact obligation, the projected index flow for the rest of the year, and the time of year with an adjustment for hydrologic losses. The Conejos Compact Curtailment percentage is calculated and stored in the CompactCalculations Data Object.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule sets Slot Inflows on the ConejosRiverBelowPlatoro object to zero. All releases from any of the conservation storage accounts (Direct Flow Storage, Relinquished Water, or Conejos Project Storage) are assigned to AllocatableFlow account on the ConejosRiverBelowPlatoro object, as this water was stored for use by diverters downstream in Colorado. Any additional release than that specified to be coming from any of the conservation accounts is split between CompactDelivery and AllocatableFlow based on the Conejos Compact Curtailment percentage.

This rule fires if it has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.  
Model slots written by rule:  
1. ConejosRiverBelowPlatoro^CompactDelivery.Slot Inflow  
2. ConejosRiverBelowPlatoro^RioGrandeAllocatableFlow.Slot Inflow  
3. RelinquishedWaterPlatoroToRioGrandeAllocatableFlowConejosRiverBelowPlatoro.Supply  
4. ConejosProjectStoragePlatoroToRioGrandeAllocatableFlowConejosRiverBelowPlatoro.Supply  
5. DirectFlowStoragePlatoroToRioGrandeAllocatableFlowConejosRiverBelowPlatoro.Supply  
6. ConejosPlatoroToCompactDeliveryConejosRiverBelowPlatoro.Supply  
7. ConejosPlatoroToRioGrandeAllocatableFlowConejosRiverBelowPlatoro.Supply  
List of key model objects with slots read by the rule or child functions:  
1. CompactCalculations  
2. ComputedDeliveries  
3. Platoro  
Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
2017: Jesse Roach. Modified the rule again with the addition of the Direct Flow Storage account to Platoro.  
2015: Jesse Roach. Modified the rule with the creation of new conservation storage accounts in Platoro.  
~2013: Craig Boroughs Created the rule.

Model slots written by rule:  
1. ConejosRiverBelowPlatoro^CompactDelivery.Slot Inflow  
2. ConejosRiverBelowPlatoro^RioGrandeAllocatableFlow.Slot Inflow  
3. RelinquishedWaterPlatoroToRioGrandeAllocatableFlowConejosRiverBelowPlatoro.Supply  
4. ConejosProjectStoragePlatoroToRioGrandeAllocatableFlowConejosRiverBelowPlatoro.Supply  
5. DirectFlowStoragePlatoroToRioGrandeAllocatableFlowConejosRiverBelowPlatoro.Supply  
6. ConejosPlatoroToCompactDeliveryConejosRiverBelowPlatoro.Supply  
7. ConejosPlatoroToRioGrandeAllocatableFlowConejosRiverBelowPlatoro.Supply

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
7/03/19: MCB. Removed disabled assignments to ConjeosProjectStoragePlatoroToConejosPlatoro.Supply and RelinquishedWaterPlatorotoConejosPlatoro.Supply  
2/13/19: Jesse Roach. Added description field and updated notes field.  
2017: Jesse Roach. Modified the rule again with the addition of the Direct Flow Storage account to Platoro.  
2015: Jesse Roach. Modified the rule with the creation of new conservation storage accounts in Platoro.  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon5.2 RPL Object IconSetRioGrandePassthroughAccountsFromUpstreamGages

Rule Purpose:  
This rule sets the passthrough account slot inflows to the upstream-most reach of the Rio Grande. There are two passthrough accounts on this reach: (1) CompactDelivery (i.e, the water allocated to meet compact delivery requirements at the Colorado-New Mexico Stateline) and (2) Rio Grande AllocatableFlow (i.e., the water that can be allocated to water users along the Rio Grande). This rule sets the slot inflows based on flow at the gages immediately upstream of the reach and the Rio Grande Compact Curtailment percentage. The Compact Curtailment percentage is calculated based on among other things, the cumulative departure of the delivery to the Colorado-New Mexico stateline from the Compact obligation, the projected index flow for the rest of the year, and the time of year with an adjustment for hydrologic losses. The Compact Curtailment percentage is calculated and stored in the CompactCalculations Data Object.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule assigns a portion of both ThirtyMileBridge and NorthClearCreekBelowContinentalReservoir inflows to the CompactDelivery account by multiplying the total inflows to each object by the CompactCalculations.RioGrandeCompactCurtailmentPercentage slot value in the previous timestep (the slot is not updated until the end of the timestep) and assigning the results to the Slot Inflows for the CompactDelivery account on the RioGrandeClearCreek Confluence object. The remainder of the inflows are assigned to the AllocatableFlow account by multiplying the total inflows by 1 minus the CompactCalculations.RioGrandeCompactCurtailmentPercentage slot value in the previous timestep, and assigning this to the Slot Inflows for the AllocatableFlow account on the RioGrandeClearCreek object. An analogous process is used to assign CompactDelivery and AllocatableFlow account Slot Inflows on the SouthForkRioGrandeConfluence object based on SouthFork inflows.

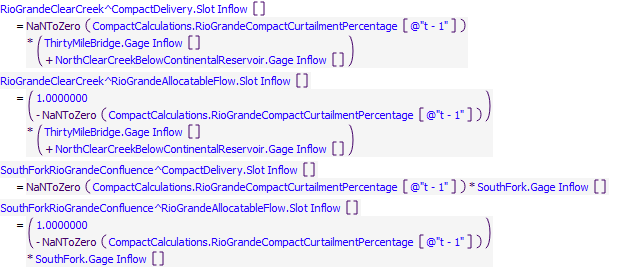
This rule fires if it has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. RioGrandeClearCreek^CompactDelivery.Slot Inflow  
2. RioGrandeClearCreek^RioGrandeAllocatableFlow.Slot Inflow  
3. SouthForkRioGrandeConfluence^CompactDelivery.Slot Inflow  
4. SouthForkRioGrandeConfluence^RioGrandeAllocatableFlow.Slot Inflow

List of key model objects with slots read by the rule or child functions:  
1. CompactCalculations  
2. NorthClearCreekBelowContientalReservoir  
3. SouthFork  
4. ThirtyMileBridge

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon5.3 RPL Object IconSetUpstreamRioSanAntonioRioLosPinosPassthroughAccounts

Rule Purpose:  
This rule sets the passthrough account slot inflows to the upstream-most gages of the of the San Antonio River (RioLosPinosAtOrtiz and RioSanAntonioAtOrtiz). There are two passthrough accounts on this reach: (1) CompactDelivery (i.e, the water allocated to meet compact delivery requirements at the Colorado-New Mexico Stateline) and (2) AllocatableFlow (i.e., the water that can be allocated to water users along the San Antonio and Conejos Rivers). This rule sets the slot inflows on the RioLosPinosAtOrtiz and RioSanAntonioAtOrtiz Reach Objects based on flow at those gages and the Conejos Compact Curtailment percentage. The Conejos Compact Curtailment percentage is calculated based on among other things, the cumulative departure of the delivery to the Colorado-New Mexico stateline from the Compact obligation, the projected index flow for the rest of the year, and the time of year with an adjustment for hydrologic losses. The Compact Curtailment percentage is calculated and stored in the CompactCalculations Data Object.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule assigns a portion of RioLosPinosAtOrtiz inflows to the CompactDelivery account by multiplying the total inflows by the CompactCalculations.ConejosCompactCurtailmentPercentage slot value in the previous timestep (the slot is not updated until the end of the timestep) and assigning the result to the Slot Inflows for the CompactDelivery account on the RioLosPinosAtOrtiz object. The remainder of the inflows are assigned to the AllocatableFlow account by multiplying the total inflows by 1 minus the CompactCalculations.ConejosCompactCurtailmentPercentage slot value in the previous timestep, and assigning this to the Slot Inflows for the AllocatableFlow account on the RioLosPinosAtOrtiz object. A analogous process is used to assign CompactDelivery and AllocatableFlow account Slot Inflows on the RioSanAntonioAtOrtiz object based on RioLosPinosAtOrtiz inflows.

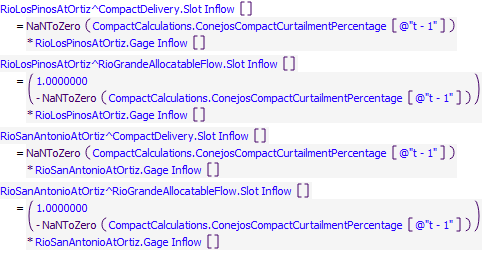
This rule fires if it has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function.

Model slots written by rule:  
1. RioLosPinosAtOrtiz^CompactDelivery.Slot Inflow  
2. RioLosPinosAtOrtiz^RioGrandeAllocatableFlow.Slot Inflow  
3. RioSanAntonioAtOrtiz^CompactDelivery.Slot Inflow  
4. RioSanAntonioAtOrtiz^RioGrandeAllocatableFlow.Slot Inflow

List of key model objects with slots read by the rule or child functions:  
1. CompactCalculations  
2. RioLosPinosAtOrtiz  
3. RioSanAntonioAtOrtiz

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon5.4 RPL Object IconSetConejosAccountsToZeroForNonIrrigationSeason

Rule Purpose:  
This rule sets all the supplies to all the diversion accounts along the Los Pinos, San Antonio, and Conejos Rivers to zero during non-irrigation season in Colorado (November through March).

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule loops through all Conejos Water Users, and for each Water User calculates the downstream lag for that user, and then loops through all accounts on the Water User, and loops through all timesteps from the current timestep to the current timestep plus the lag and sets all diversion account supplies during that time period to zero.

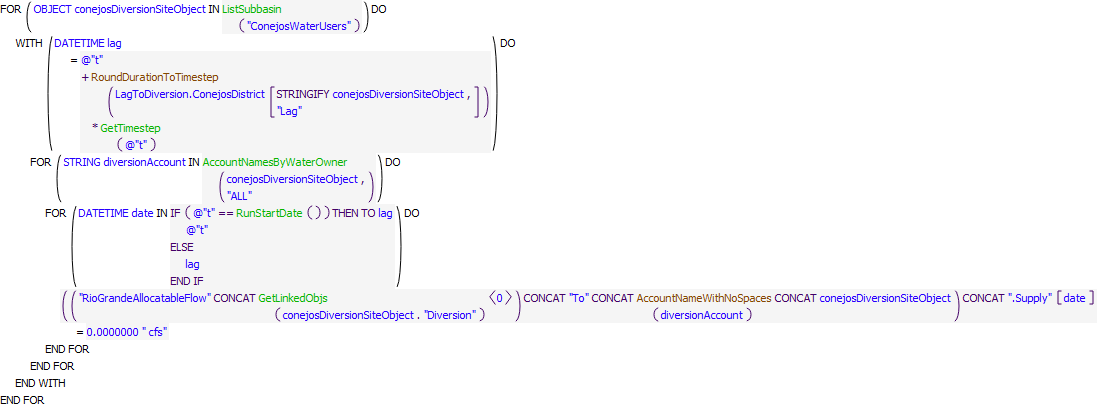
This rule fires once each timestep when it is not irrigation season.

Model slots written by rule:  
1. A total of 190 slots on the Conejos system.

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
11/17/2017: Jesse Roach. Updated to make general to daily or monthly timestep  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**IsConejosDistrict22IrrigationSeason
* **RPL Object Icon**AccountNameWithNoSpaces
* **RPL Object Icon**RoundDurationToTimestep
* **RPL Object Icon**GetTimestep
* **RPL Object Icon**RunStartDate
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**GetLinkedObjs
* **RPL Object Icon**AccountNamesByWaterOwner
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon5.5 RPL Object IconSetRioGrandeAccountsToZeroForNonIrrigationSeason

Rule Purpose:  
This rule sets all the supplies to all the diversion accounts along the Rio Grande in Colorado to zero during non-irrigation season in Colorado (November through March).

Rule Logic: Execution Constraint logic is at end of explanation.This rule loops through all Rio Grande Water Users, and for each Water User calculates the downstream lag for that user, and then loops through all accounts on the Water User, and loops through all timesteps from the current timestep to the current timestep plus the lag and sets all diversion account supplies during that time period to zero.

This rule fires if it has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND if the timestep is outside the Conejos District 22 Irrigation Season (April 1 through October 31)

Model slots written by rule:  
1. A total of 298 slots in the Rio Grande system.

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
11/17/2017: Jesse Roach. Updated to make general to daily or monthly timestep  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**IsConejosDistrict22IrrigationSeason
* **RPL Object Icon**AccountNameWithNoSpaces
* **RPL Object Icon**RoundDurationToTimestep
* **RPL Object Icon**GetTimestep
* **RPL Object Icon**RunStartDate
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**GetLinkedObjs
* **RPL Object Icon**AccountNamesByWaterOwner
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon5.6 RPL Object IconSetConejosDiversionAccounts

Rule Purpose:  
This rule sets the supplies to all the Conejos system diversion accounts in Colorado during the irrigation season from the chain of Rio Grande Allocatable Flow accounts. Supplies are allocated first to meet the needs of irrigators with a senior priority date (i.e., the oldest priority date has the highest priority). When there is enough water in the Rio Grande Allocatable Flow supply chain to meet the demand of all water users, then all water users will receive their full water right. When there is not enough water in the Rio Grande Allocatable Flow supply chain to meet this demand, some users will be shorted based on priority.

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule allocates based on water rights by calling the RiverWare function SolveWaterRightsWithLags for the ConejosDistrict computational sub-basin with ConejosAllocatable water as the available water for allocation.

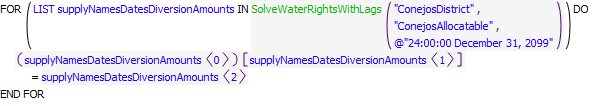
This rule fires if it has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND if the timestep is within the Conejos District 22 Irrigation Season (April 1 through October 31)

Model slots written by rule:  
1. A total of 190 slots in the Conejos System

List of key model objects with slots read by the rule or child functions:  
1. Objects in the ConejosDistrict computational subbasin.

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**IsConejosDistrict22IrrigationSeason
* **RPL Object Icon**IsHistoricalModelRun
* **RPL Object Icon**SolveWaterRightsWithLags
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon5.7 RPL Object IconSetRioGrandeDiversionAccounts

Rule Purpose:  
This rule sets the supplies to all the Rio Grande system diversion accounts in Colorado during the irrigation season from the chain of Rio Grande Allocatable Flow accounts. Supplies are allocated first to meet the needs of irrigators with a senior priority date (i.e., the oldest priority date has the highest priority). When there is enough water in the Rio Grande Allocatable Flow supply chain to meet the demand of all water users, then all water users will receive their full water right. When there is not enough water in the Rio Grande Allocatable Flow supply chain to meet this demand, some users will be shorted based on priority.

Rule Logic: Execution Constraint logic is at end of explanation.This rule allocates based on water rights by calling the RiverWare function SolveWaterRightsWithLags for the RioGrandeDistrict computational sub-basin with RioGrandeAllocatableFlow water as the available water for allocation.

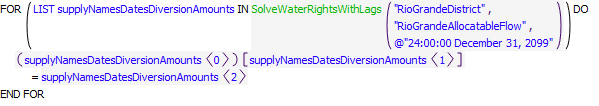
This rule fires if it has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function AND if the timestep is within the Rio Grange District 20 Irrigation Season (April 1 through October 31)

Model slots written by rule:  
1. RioGrandeAllocatableFlow"Reach Object"To"Account""Water User Object".Supply for all "Accounts" on all "Water User Objects" diverting from linked "Reach Objects" on the Rio Grande in Colorado. This is a total of 298 slots in the Rio Grande system.

List of key model objects with slots read by the rule or child functions:  
1. Objects in the RioGrandeDistrict computational subbasin.

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**IsRioGrandeDistrict20IrrigationSeason
* **RPL Object Icon**IsHistoricalModelRun
* **RPL Object Icon**SolveWaterRightsWithLags
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon5.8 RPL Object IconSetConejosPhysicalDiversionRequestedAmounts

Rule Purpose:  
This rule sets both the physical diversion requested, and depletions requested on the diversion sites along the Los Pinos, San Antonio, and Conejos Rivers equal to the accounting supplies set previously by the lower priority accounting based rules. The timestep set on each diversion object slot is the timestep the water at the top of the Conejos computational subbasin on the current timestep will arrive at each diversion (lagged date)

Rule Logic: Execution Constraint logic is at end of explanation.  
This rule uses an outer FOR loop to loop through all diversion objects in the ConejosWaterUsers subbasin. Within this loop, a WITH loop sets the lag data for each diversion object (the date the water at the top of the basin in the current timestep will arrive at the diversion). The third loop in the rule, internal to both of the others, is a FOR loop that handles the special case of the first timestep to fill in all requests at lagged locations for water that entered the basin before the start of the model. Within these three loops, the "Diversion Requested" slot is set on each diversion object with a FOR loop that sums the water rights based requests on all account supplies associated with the given diversion object (as set previously in the timestep by the Water Rights Solver within irrigation season, or to zero outside irrigation season). The "Depletion Requested" is set on each diversion object by multiplying one less the return flow percentage associated with each diversion object by the diversion requested. This depletion requested calls an identical FOR loop to duplicate the diversion requested result.

This rule fires if it has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function  
Comments:  
Some sort of efficiency gain seems possible with respect to the identical FOR loops, but after several attempts it is not obvious how to achieve this gain.

Model slots written by rule:  
1. The conejosDiversionSiteObject.Diversion Requested slot on all 114 Diversion Objects diverting from the Conejos, Los Pinos, or San Antonio rivers in Colorado.  
2. The conejosDiversionSiteObject.Depletion Requested slot on all 114 Diversion Objects diverting from the Conejos, Los Pinos, or San Antonio rivers in Colorado.

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/16/18: Jesse Roach. Failed attempt to make the rule more efficient.  
11/17/17: Jesse Roach. Updated to make general to daily or monthly timestep  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**AccountNameWithNoSpaces
* **RPL Object Icon**RoundDurationToTimestep
* **RPL Object Icon**GetTimestep
* **RPL Object Icon**RunStartDate
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**GetLinkedObjs
* **RPL Object Icon**AccountNamesByWaterOwner
* **RPL Object Icon**HasRuleFiredSuccessfully

### Control Display Icon5.9 RPL Object IconSetRioGrandePhysicalDiversionRequestedAmounts

Rule Purpose:  
This rule sets both the physical diversion requested, and depletions requested on the diversion sites along the Rio Grande in Colorado equal to the accounting supplies set previously by the lower priority accounting based rules. The timestep set on each diversion object slot is the timestep the water at the top of the Rio Grande computational subbasin on the current timestep will arrive at each diversion (lagged date).

Rule Logic: Execution Constraint logic is at end of explanation.This rule uses an outer FOR loop to loop through all diversion objects in theRioGrandeWaterUsers subbasin. Within this loop, a WITH loop sets the lag data for each diversion object (the date the water at the top of the basin in the current timestep will arrive at the diversion). The third loop in the rule, internal to both of the others, is a FOR loop that handles the special case of the first timestep to fill in all requests at lagged locations for water that entered the basin before the start of the model. Within these three loops, the "Diversion Requested" slot is set on each diversion object with a FOR loop that sums the water rights based requests on all account supplies associated with the given diversion object (as set previously in the timestep by the Water Rights Solver within irrigation season, or to zero outside irrigation season). The "Depletion Requested" is set on each diversion object by multiplying one less the return flow percentage associated with each diversion object by the diversion requested. This depletion requested calls an identical FOR loop to duplicate the diversion requested result.

This rule fires if it has not fired yet as checked with reference to the predefined HasRuleFiredSuccessfully function  
Comment:  
Some sort of efficiency gain seems possible with respect to the identical FOR loops, but after several attempts it is not obvious how to achieve this gain.

Model slots written by rule:  
1. The rioGrandeDiversionSiteObject.Diversion Requested slot on all 63 Diversion Objects diverting from the Rio Grande in Colorado.  
2. The rioGrandeDiversionSiteObject. Depletion Requested slot on all 63 Diversion Objects diverting from the Rio Grande in Colorado.

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

Log of when and how the rule has been modified (newest changes at the top):  
Date: Who. What  
2/13/19: Jesse Roach. Added description field and updated notes field.  
8/16/18: Jesse Roach. Failed attempt to make the rule more efficient.  
11/17/17: Jesse Roach. Updated to make general to daily or monthly timestep  
~2013: Craig Boroughs Created the rule.

Statements

****

Execution Constraint

**Execution Constraint**

Referenced Functions

* **RPL Object Icon**AccountNameWithNoSpaces
* **RPL Object Icon**RoundDurationToTimestep
* **RPL Object Icon**GetTimestep
* **RPL Object Icon**RunStartDate
* **RPL Object Icon**ListSubbasin
* **RPL Object Icon**GetLinkedObjs
* **RPL Object Icon**AccountNamesByWaterOwner
* **RPL Object Icon**HasRuleFiredSuccessfully