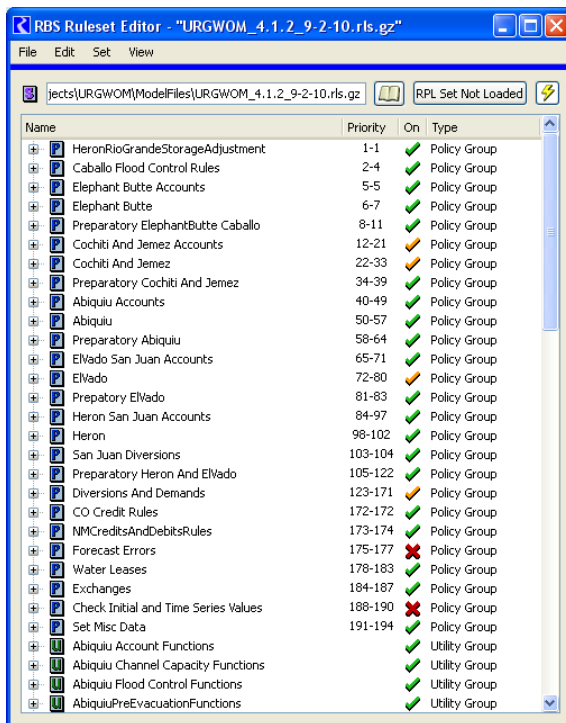


# RiverWare Ruleset Documentation

Version 4.1.2

## Upper Rio Grande Water Operations Model (URGWOM)



The screenshot shows the RBS Ruleset Editor window titled "URGWOM\_4.1.2\_9-2-10.rls.gz". The window displays a list of ruleset items with columns for Name, Priority, On, and Type. The items are organized into a tree structure with expandable/collapsible icons. The 'On' column contains green checkmarks for most items, orange checkmarks for some, and red X marks for others. The 'Type' column lists various categories like Policy Group and Utility Group.

Name	Priority	On	Type
HeronRioGrandeStorageAdjustment	1-1	✓	Policy Group
Caballo Flood Control Rules	2-4	✓	Policy Group
Elephant Butte Accounts	5-5	✓	Policy Group
Elephant Butte	6-7	✓	Policy Group
Preparatory ElephantButte Caballo	8-11	✓	Policy Group
Cochiti And Jemez Accounts	12-21	✓	Policy Group
Cochiti And Jemez	22-33	✓	Policy Group
Preparatory Cochiti And Jemez	34-39	✓	Policy Group
Abiquiu Accounts	40-49	✓	Policy Group
Abiquiu	50-57	✓	Policy Group
Preparatory Abiquiu	58-64	✓	Policy Group
ElVado San Juan Accounts	65-71	✓	Policy Group
ElVado	72-80	✓	Policy Group
Preparatory ElVado	81-83	✓	Policy Group
Heron San Juan Accounts	84-97	✓	Policy Group
Heron	98-102	✓	Policy Group
San Juan Diversions	103-104	✓	Policy Group
Preparatory Heron And ElVado	105-122	✓	Policy Group
Diversions And Demands	123-171	✓	Policy Group
CO Credit Rules	172-172	✓	Policy Group
NMCreditsAndDebitsRules	173-174	✓	Policy Group
Forecast Errors	175-177	✗	Policy Group
Water Leases	178-183	✓	Policy Group
Exchanges	184-187	✓	Policy Group
Check Initial and Time Series Values	188-190	✗	Policy Group
Set Misc Data	191-194	✓	Policy Group
Abiquiu Account Functions		✓	Utility Group
Abiquiu Channel Capacity Functions		✓	Utility Group
Abiquiu Flood Control Functions		✓	Utility Group
AbiquiuPreEvacuationFunctions		✓	Utility Group

*prepared by*

Craig Boroughs, Ph.D., P.E.

September 2010

# TABLE OF CONTENTS

EXECUTIVE SUMMARY .....	viii
I. Introduction .....	1
1.1. URGWOM Ruleset.....	2
II. Synopses of Policy Controls .....	4
2.1. San Juan-Chama Project.....	4
2.2. Flood Control Operations .....	8
2.3. Compact.....	10
2.3.1. Article VII of the Compact.....	12
2.3.2. Relinquished Credits and Emergency Drought Water .....	12
2.4. Diversions .....	13
2.4.1. Rio Chama Diversions/Acequias .....	13
2.4.2. MRGCD Diversions .....	13
2.4.3. Six Middle Valley Pueblos.....	14
2.4.3.1. Increased Angostura Diversions .....	14
2.4.4. Albuquerque Diversion .....	14
2.5. Downstream Target Flows.....	15
2.5.1. Hydrology Year Type.....	15
2.5.2. Adjustment Factor .....	16
2.5.3. Discretionary Operations.....	16
2.5.4. Supplemental Water Needed For Targets .....	16
2.5.5. Shorted Diversions .....	19
2.5.6. Pumping from the Low Flow Conveyance Channel .....	19
2.6. Recreation .....	20
III. Priorities for Rule Execution .....	21
3.1. Heron .....	22
3.2. El Vado.....	23
3.3. Abiquiu .....	23
3.4. Cochiti and Jemez.....	24
3.4.1. Cochiti Deviations.....	25
3.5. Elephant Butte and Caballo .....	26
IV. Other Topics .....	27
4.1. Planning Model versus Water Operations Model.....	27
4.2. User-defined vs. Predefined Functions.....	28
4.3. Accounting.....	28
4.3.1. Supplies .....	28
4.3.2. Exchanges.....	29
4.4. Other RiverWare Features .....	29
V. Conclusions.....	31
REFERENCES .....	31

## APPENDIX A

A. Individual Rules .....	A-1
A.1. Set Misc. Data.....	A-1
A.1.1. Set Supply NaNs To Zero .....	A-1
A.1.2. Set Average Nambe Falls Data for RGCompact.....	A-2
A.1.3. SetCombinedDeliveryRequest .....	A-3
A.1.4. SetAlbuquerqueDeliveryRequest .....	A-7
A.2. Check Initial and Time Series Values .....	A-10
A.2.1. Check Model Inputs Temperature .....	A-11
A.2.2. Check Model Inputs Pan Evaporation.....	A-11
A.2.3. CheckModelInputsWithListFunctions .....	A-12
A.3. Exchanges .....	A-13
A.3.1. SetAlbuquerqueJemezEXs .....	A-13
A.3.2. SetAlbuquerqueLoanEXs.....	A-13
A.3.3. SetNoAlbuquerqueLoanEXs .....	A-14
A.3.4. SetHeronElVadoEXs .....	A-15
A.4. Water Leases.....	A-16
A.4.1. Heron Lease Amounts .....	A-16
A.4.2. El Vado Lease Amounts.....	A-17
A.4.3. Abiquiu Lease Amounts.....	A-17
A.4.4. Set MRGCD Loans .....	A-20
A.4.5. Set Reclamation ElVado Abiquiu Payback.....	A-23
A.4.6. Check Delivery Requests Input.....	A-25
A.5. Forecast Errors.....	A-25
A.5.1. RewindRandomFile.....	A-26
A.5.2. CalculatedForecastError .....	A-26
A.5.3. ForecastErrorPercent .....	A-27
A.6. NMCreditsAndDebitsRules .....	A-29
A.6.1. Set NM Credit Adjustment.....	A-29
A.6.2. Zero NM CO Credits .....	A-29
A.7. CO Credit Rules.....	A-30
A.7.1. CO Credit .....	A-30
A.8. Diversions and Demands .....	A-31
A.8.1. SanMarcialElephantButteRiparianLoss .....	A-31
A.8.2. SanAcaciaSocorroMainCanalDiversionRequest.....	A-31
A.8.3. SanAcaciaLFCCDiversionRequest .....	A-32
A.8.4. CentralWastewayCalcs.....	A-33
A.8.5. LFCCPumpingRequested.....	A-33
A.8.6. CompactVIIUsableStorage.....	A-34
A.8.7. SetCompactVIISwitch.....	A-36
A.8.8. ComputeOtowiForecast.....	A-37
A.8.9. Hydrology Year Type.....	A-37
A.8.10. SetMinTargetsAtStart.....	A-38
A.8.11. MinIsletaSanAcaciaSanMarcialFlowTargets .....	A-39
A.8.12. MinCentralFlow Target.....	A-40
A.8.13. ResetIsletaSanAcaciaSanMarcialTargetsForStepDown .....	A-41

A.8.14. WriteNewMinTargetsForOverbankHydrographKnowledge.....	A-42
A.8.15. WriteNewMinTargetsForRecruitmentHydrographKnowledge.....	A-44
A.8.16. ComputeCochitiRGConservationSpaceAvailable.....	A-45
A.8.17. EndTargetsForOverbankOrRecruitment .....	A-47
A.8.18. SetRioChamaDiversionAndDepletionRequested.....	A-48
A.8.19. SetMiddleRioGrandeDiversionRequested .....	A-50
A.8.20. ResetAngosturaDiversionForShortageOps .....	A-50
A.8.21. SetPreemptiveAlbuquerqueCutoffSwitch .....	A-51
A.8.22. SetAlbuquerqueDiversion .....	A-52
A.8.23. CheckPreemptiveCutoffVersusPermitTEST .....	A-55
A.8.24. InitialEstimateElVadoAndAbiquiuAndCochitiInflow .....	A-55
A.8.25. Set Estimated ElVado Inflow .....	A-57
A.8.26. Set Estimated Abiquiu Inflow .....	A-58
A.8.27. Set Estimated Cochiti Inflow .....	A-59
A.8.28. InitialFlowToMeetCentralAndIsletaAndSanAcaciaAndSanMarcial .....	A-59
A.8.29. Shorted Middle Valley Diversions .....	A-60
A.8.29.1. CochitiReleaseShortingMiddleValleyDiversions .....	A-61
A.8.30. ComputeReleaseToMeetMinimumCentralFlow .....	A-62
A.8.31. ComputeReleaseToMeetMinimumIsletaFlow .....	A-63
A.8.32. ComputeReleaseToMeetMinimumSanAcaciaFlow .....	A-63
A.8.33. ComputeReleaseToMeetMinimumSanMarcialFlow.....	A-63
A.8.34. SetCochitiMinimumFlow .....	A-63
A.8.35. ComputeAbiquiuMRGCDDemand .....	A-64
A.8.36. SetAbiquiuMRGCDDemand.....	A-64
A.8.37. SetRaftingRelease .....	A-65
A.8.38. SetElVadoMRGCDDemand .....	A-66
A.8.39. Indian Storage Requirement Release First or Second Timestep .....	A-66
A.8.40. Indian Storage Requirement Release .....	A-67
A.8.41. SetMinElVadoMRGCDDemand.....	A-69
A.8.42. ComputedMinAbiquiuRGRelease.....	A-69
A.8.43. SetMinAbiquiuRGRelease .....	A-70
A.8.44. AbiquiuTotalFlowToMeetTarget .....	A-70
A.9. Preparatory Heron And ElVado.....	A-71
A.9.1. SetRelinquishedNMCredits.....	A-71
A.9.2. UpdateEmergencyDroughtStorageAllocations .....	A-71
A.9.3. SetCumulativeCochitRecPoolRelease.....	A-72
A.9.4. SetCumulativeAccountFillRelease.....	A-73
A.9.5. SetSanJuanContractorAllocationsJuly1 .....	A-74
A.9.6. SetSanJuanContractorAllocations .....	A-75
A.9.7. SetPercentRGRelease .....	A-77
A.9.7.1 TargetFillDate .....	A-78
A.9.7.2. ElVadoAccountStorageAdjustment .....	A-79
A.9.7.3. ElVadoIsPriority .....	A-80
A.9.7.4. CurrentYearIsWaiverYear .....	A-80
A.9.8. SetElVadoIndianStorageReqAprilAndMarch .....	A-82
A.9.8.1. ComputeSupplyAtOtowi.....	A-83

A.9.9. SetElVadoIndianStorageReqAfterApril .....	A-84
A.9.10. ZeroElVadoIndianStorageReq .....	A-85
A.9.11. ComputeHeronRGRelease .....	A-85
A.9.12. ElVadoIndianAccountTransfer .....	A-89
A.9.13. SetElVadoIndianStorageInflow.....	A-90
A.9.14. SetInflowToMRGCDDroughtStorage.....	A-91
A.9.15. SetInflowToSupplementalESASStorage .....	A-92
A.9.16. ComputeElVadoRGRelease .....	A-92
A.9.17 EstimateHeronSJRelease.....	A-95
A.9.18. SetCarryover.....	A-96
A.10. San Juan Diversions.....	A-98
A.10.1. ComputeAnnualSJDiversions .....	A-98
A.10.2. San Juan Diversions .....	A-98
A.11. Heron .....	A-101
A.11.1. HeronSJReleaseRestrictions.....	A-101
A.11.2. HeronOutflow.....	A-101
A.11.2.1. CheckThisResPhysicalConstraints.....	A-102
A.11.3. HeronCheckDeltaStorage.....	A-104
A.11.4. SetHeronRGAccount.....	A-104
A.11.5. CheckHeronRGOutflow .....	A-107
A.12. Heron San Juan Accounts .....	A-108
A.12.1. SetHeronCochitiRecPoolAccounts .....	A-108
A.12.1.1. RuleIsInPriority.....	A-109
A.12.2. SetHeronMRGCDDPaybacksAbiquiuAccounts.....	A-109
A.12.3. SetHeronMRGCDDPaybacksElVadoAccounts.....	A-110
A.12.4. SetHeronElVadoDeliveryAccounts.....	A-110
A.12.5. SetHeronNMISCAccounts .....	A-111
A.12.6. SetHeronOtowipaybacksAccounts.....	A-111
A.12.7. SetHeronAlbuquerquePaybacksAccounts.....	A-111
A.12.8. SetHeronWaiverAbiquiuAccounts.....	A-112
A.12.9. SetHeronWaiverElVadoAccounts.....	A-113
A.12.10. SetHeronAccountFillAbiquiuAccounts.....	A-113
A.12.11. SetHeronAccountFillElVadoAccounts .....	A-114
A.12.12. SetHeronAccountDeliveryAbiquiuAccounts .....	A-115
A.12.13. SetHeronAccountDeliveryElVadoAccounts .....	A-115
A.12.14. SetHeronReleaseToMRGCDAccounts .....	A-115
A.13. Preparatory ElVado .....	A-116
A.13.1. ComputeElVadoSJRelease.....	A-116
A.13.2. SetWaiverBalance .....	A-118
A.13.3. ZeroWaivers .....	A-119
A.14. ElVado .....	A-119
A.14.1. ElVadoOutflow .....	A-120
A.14.2. ElVadoRafting.....	A-120
A.14.3. ElVadoIndianPuebloWaterRight.....	A-121
A.14.4. ElVadoFloodControl .....	A-122
A.14.5. ElVadoChannelCapacity .....	A-122

A.14.6. SetRioGrandeAccountCompactVII.....	A-123
A.14.7. SetRemainingElVadoRGAccountsCompactVII .....	A-125
A.14.8. SetElVadoRGAccounts .....	A-125
A.14.9. CheckElVadoRGOutflow.....	A-126
A.15. ElVado San Juan Accounts.....	A-126
A.15.1. SetElVadoFlowThroughAccounts.....	A-127
A.15.1.1. RuleIsInPriority.....	A-128
A.15.2. SetElVadoMRGCDAccount .....	A-128
A.15.3. SetElVadoReclamationAccount.....	A-129
A.15.4. SetElVadoOtowipaybacksAccounts .....	A-130
A.15.5. SetElVadoAlbuquerquePaybacksAccounts.....	A-130
A.15.6. SetElVadoMRGCDPaybacksAccounts.....	A-130
A.15.7. SetElVadoAccountFillAccounts .....	A-130
A.16. Preparatory Abiquiu.....	A-131
A.16.1. SetRioGrandeToReclamationTransfer .....	A-131
A.16.2. AbiquiuLockedIn.....	A-132
A.16.3. ComputeAbiquiuRGRelease .....	A-133
A.16.4. ComputeAbiquiuMinFlowsDemand .....	A-134
A.16.5. SetAbiquiuMinFlowsDemand.....	A-135
A.16.6. SetAlbuquerqueLoanAccounts.....	A-136
A.16.7. ComputeAbiquiuSJRelease .....	A-137
A.17. Abiquiu .....	A-138
A.17.1. Abiquiu Outflow.....	A-138
A.17.2. AbiquiuMinimumFlow.....	A-140
A.17.3. AbiquiuPreEvacuation.....	A-140
A.17.4. AbiquiuSteppedRelease.....	A-142
A.17.5. AbiquiuChannelCapacityRestrictions .....	A-145
A.17.6. AbiquiuTemporaryFlowsForMaintenance .....	A-146
A.17.7. AbiquiuFloodControl .....	A-146
A.17.8. AbiquiuRGCarryOver .....	A-147
A.18. Abiquiu Accounts .....	A-147
A.18.1. SetAbiquiuRGConservationAccount .....	A-148
A.18.2. SetAbiquiuRGAccount.....	A-149
A.18.3. CheckAbiquiuRGOutflow.....	A-150
A.18.4. SetAbiquiuFlowThroughAccounts.....	A-150
A.18.5. SetAbiquiuOtowipaybacksAccounts.....	A-151
A.18.6. SetAbiquiuMRGCDAccount.....	A-152
A.18.7. SetAbiquiuAlbuquerqueAccount .....	A-153
A.18.8. SetAbiquiuReclamationAccounts .....	A-154
A.18.9. SetAbiquiuNMISCAccounts .....	A-155
A.18.10. AlbuquerqueToRioGrandeTransfer.....	A-155
A.19. Preparatory Cochiti And Jemez .....	A-156
A.19.1. CochitiLockedIn.....	A-156
A.19.2. ComputeCochitiRGRelease.....	A-156
A.19.3. ComputeCochitiRGConservationRelease .....	A-157
A.19.4. ComputeCochitiSJRelease .....	A-158

A.19.5. ComputeJemezRGRelease .....	A-159
A.19.6. SetJemezSJRelease.....	A-159
A.20. Cochiti And Jemez.....	A-161
A.20.1. Cochiti Outflow .....	A-161
A.20.2. JemezOutflow.....	A-162
A.20.3. CentralChannelCapacityRule .....	A-163
A.20.4. SanMarcialChannelCapacityRule .....	A-164
A.20.5. JemezSanMarcialChannelCapacity .....	A-164
A.20.6. CochitiChannelCapacityRestrictions.....	A-165
A.20.7. CochitiSteppedReleaseWithCochitiDeviation .....	A-165
A.20.8. CochitiWCMBalancedRelease.....	A-166
A.20.9. CochitiFloodControl.....	A-170
A.20.10. CochitiRGCarryOver.....	A-171
A.21. Cochiti and Jemez Accounts.....	A-173
A.21.1. SetCochitiRGConservationAccountInflow .....	A-173
A.21.2. SetCochitiRGAccount .....	A-173
A.21.3. CheckCochitiRGOutflow .....	A-174
A.21.4. SetCochitiFlowThroughAccounts .....	A-174
A.21.5. SetCochitiReclamationAccounts.....	A-175
A.21.6. RGTransfer .....	A-175
A.21.7. SetJemezAccounts.....	A-176
A.21.8. CheckJemezRGOutflow .....	A-177
A.21.9. ReconciledRGandSJReleases .....	A-178
A.22. Preparatory ElephantButte Caballo .....	A-178
A.22.1. NM Relinqueshments .....	A-178
A.22.2. Determine Percentage.....	A-179
A.22.3. ElephantButte Downstream Demand .....	A-179
A.22.4. Caballo Downstream Demand.....	A-180
A.23. Elephant Butte .....	A-182
A.23.1. ElephantButteOutflow .....	A-182
A.23.2. ElephantButteOutflowRestrictions.....	A-182
A.24. Elephant Butte Accounts .....	A-184
A.24.1. SetElephantButteAccounts.....	A-184
A.25. Caballo Flood Control Rules .....	A-184
A.25.1. CaballoOutflow .....	A-184
A.25.2. CaballoOutflowRestrictions .....	A-185
A.25.3. CaballoFloodControlReleaseBasedOnFORD .....	A-186
A.26. HeronRioGrandeStorageAdjustment.....	A-187
A.26.1. HeronRioGrandeAdjustment.....	A-187

# EXECUTIVE SUMMARY

A RiverWare model for the Upper Rio Grande in New Mexico was developed with the RiverWare software application. Rulebased simulations are completed with the Upper Rio Grande Water Operations Model (URGWOM) using a RiverWare ruleset that represents policy for operating the facilities in the basin. The ruleset includes code in the RiverWare rule policy language for computing key demands, diversions, and releases from dams based on all the policy factors. The releases from the dams along the Rio Chama and Rio Grande are set for each day in the daily timestep model based on the policy coded in the ruleset. The ruleset has been under development for several years with involvement from an interagency Technical Team and contractors.

This report was prepared to provide documentation of the URGWOM RiverWare ruleset and provide new RiverWare users with a document that can be referenced when working with the ruleset. Background information on key policy controls are presented to prepare users for working with the model and to also summarize the key aspects of the policy as coded in the RiverWare ruleset. The key policy controls that are specifically discussed are the San Juan-Chama Project; flood control operations; the Compact; diversions for the Middle Rio Grande Conservancy District (and the six Middle Valley pueblos), the Albuquerque Bernalillo County Water Utility Authority, and from the Rio Chama; downstream target flows; and recreational interests. Separate discussions of priorities at each of the following dams are also presented: Heron, El Vado, Abiquiu, Cochiti, Jemez, Elephant Butte, and Caballo.

Discussion on several different aspects of operations or proposed actions that can be simulated with URGWOM is presented including relinquished Compact credits and storage of Emergency Drought water, pumping from the Low Flow Conveyance Channel to the river, and Cochiti deviations. Background information is presented on letter water deliveries by contractors to payback the river for depletions in the basin, leases of San Juan-Chama Project water by the U.S. Bureau of Reclamation, and waivers issued to contractors to allow allocated San Juan-Chama Project water to be stored at Heron Reservoir into the year following the year of the allocation.

This report does not serve as documentation of RiverWare or the functionalities of the software, but some aspects of RiverWare are discussed that are used abundantly in the model and may need to be specifically understood by new users before working with the ruleset.

All the individual rules in the current ruleset (version 4.1.2) are discussed separately in Appendix A, and flowcharts are presented to illustrate policy for some rules and aspects of operations. The appendix with discussion for individual rules makes up a significant portion of this report and was specifically prepared such that model users would have a reference when working with the RiverWare rule policy language. The discussion in this report could also be referenced as a means to complete an initial review of the policy coded in URGWOM.



# I. Introduction

The Upper Rio Grande Water Operations Model (URGWOM) is a computational model developed through an interagency effort to simulate processes and operations of facilities in the Rio Grande Basin in New Mexico. Four separate URGWOM modules are utilized for modeling exercises and decision support: Planning, Water Operations, Forecast, and Accounting. A separate URGWOM ruleset is used with the Planning and Water Operations modules. The Planning Model is used to complete daily timestep rulebased simulations for different operation scenarios to evaluate subsequent long-term impacts of a proposed action on various indicators including deliveries to water users, river flows, interstate Compact deliveries and Compact status, and the overall water budget. The Water Operations Model is used to complete daily timestep rulebased simulations to forecast operations, deliveries, and resulting flows through the end of a calendar year for Annual Operating Plans (AOP) with forecasted inflows computed using the Forecast Model. The Accounting Model is used to complete model runs, without rules, up to the current date using actual data to simulate the status of accounts for different water users.

This report provides documentation of the policy for Rio Grande operations in New Mexico as coded in the single URGWOM ruleset used with both the Water Operations Model and Planning Model. A trigger slot is included in both models that is referenced as needed within the rules to identify whether the ruleset is being used with the Water Operations Model or Planning Model. This document serves as a reference for URGWOM users working with the RiverWare ruleset and also provides a means to review the policy as coded in the model. The version number for this document matches the version number for the corresponding ruleset file. Refer to the URGWOM Quality Control Plan for more information on the protocol for model versioning.

A brief review of RiverWare rules is initially presented followed by a discussion of the policy factors affecting Rio Grande operations. A general description of the key controls that affect operations is presented. San Juan-Chama project diversions and paper accounting for San Juan water, the interstate Compact, policy for flood operations, diversions by water users including Middle Rio Grande Conservancy District (MRGCD) and the six Indian pueblos, target flows to comply with the Endangered Species Act (ESA) of 1973, and recreational interests are key driving forces that ultimately affect the outflow from each dam along the river. A general discussion of these factors is presented to summarize the key aspects that affect policy and prepare model users for working with URGWOM. The priorities for operating each of the major dams are then discussed separately.

While this report does not serve as documentation of the RiverWare software, some particular aspects of the software are discussed in this report. A specific understanding of these topics may be important before new users begin working with the URGWOM ruleset.

A more detailed summary for each individual rule is presented with a brief explanation of the assignments made with each rule, a discussion of the execution constraints, and a

summary of the rule logic. Flow charts are also presented that depict the rule logic for individual rules. All the rules in the current ruleset including those specifically designated for either the water operations model or the planning model are discussed in this report. Older or temporary rules that are turned off are also included to assure the list of rules is comprehensive. The level of detail for the summaries varies depending on the complexity of the rule, and discussion may not be repeated for rules that contain identical logic.

## 1.1. URGWOM Ruleset

The URGWOM model was developed with the RiverWare software application developed at the Center for Advanced Decision Support for Water and Environmental Systems (CADSWES) at the University of Colorado at Boulder. The model is a daily timestep model for simulating operations of facilities along the Rio Grande (and Rio Chama) within New Mexico including Heron, El Vado, Abiquiu, Cochiti, Jemez, Elephant Butte, and Caballo Dams. Diversions at numerous locations along the Rio Chama and Rio Grande are simulated along with drain and canal flows in the Middle Valley between Cochiti Dam and Elephant Butte Reservoir. Refer to Figure 1.1 for a map of the Rio Grande basin in New Mexico.

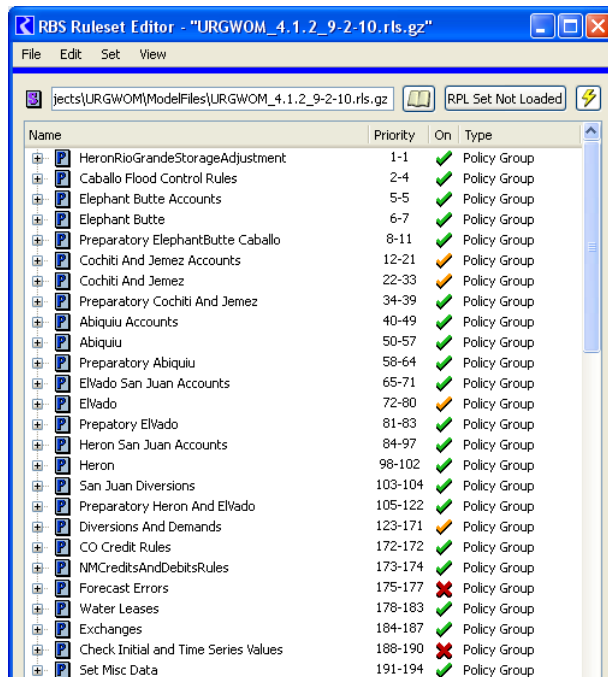


Figure 1.1. Map of the Rio Grande Basin (courtesy of the Upper Rio Grande Water Operations Review)

URGWOM has been under development for many years and changes are always being made to enhance the model. This report focuses solely on the ruleset portion of the model which contains code for completing daily timestep rulebased simulations. The ruleset includes a series of rules coded in the RiverWare rule policy language using the graphical rules editor in RiverWare. The series of rules represents the policy for operating the facilities throughout the basin in New Mexico and ultimately setting the daily outflows from each dam.

For URGWOM version 4.1.2, there are 194 rules coded which represent the policy in the basin for such objectives as setting diversions, recording paper water transfers, meeting downstream target flows, adhering to flood control operations criteria, all to ultimately determine the daily outflow from each dam in the basin. (Note that a few old rules are turned off but still included in the ruleset. Rules that are turned off and do not execute are noted with a red X in the ruleset. Such rules are usually under development or older rules that have been turned off but retained as new rules are tested. As new rules are deemed appropriate, the older rules or temporary rules that were turned off are deleted.)

The rules are sorted in 26 different policy groups (Refer to Figure 1.2). Assignments for the different parameters as completed with each rule are based on the priority for the rule. For example, setting an outflow based on flood control policy will have a higher priority than a delivery to water users. The rules are listed based on priority and are executed, or fired, in reverse order. In addition to the priority of the policy, the rules set outflows from the dams from upstream to downstream. Please refer to documentation on RiverWare for more details on rulebased simulations.



Name	Priority	On	Type
HeronRioGrandeStorageAdjustment	1-1	✓	Policy Group
Caballo Flood Control Rules	2-4	✓	Policy Group
Elephant Butte Accounts	5-5	✓	Policy Group
Elephant Butte	6-7	✓	Policy Group
Preparatory ElephantButte Caballo	8-11	✓	Policy Group
Cochiti And Jemez Accounts	12-21	✓	Policy Group
Cochiti And Jemez	22-33	✓	Policy Group
Preparatory Cochiti And Jemez	34-39	✓	Policy Group
Abiquiu Accounts	40-49	✓	Policy Group
Abiquiu	50-57	✓	Policy Group
Preparatory Abiquiu	58-64	✓	Policy Group
ElVado San Juan Accounts	65-71	✓	Policy Group
ElVado	72-80	✓	Policy Group
Preparatory ElVado	81-83	✓	Policy Group
Heron San Juan Accounts	84-97	✓	Policy Group
Heron	98-102	✓	Policy Group
San Juan Diversions	103-104	✓	Policy Group
Preparatory Heron And ElVado	105-122	✓	Policy Group
Diversions And Demands	123-171	✓	Policy Group
CO Credit Rules	172-172	✓	Policy Group
NMCreditsAndDebitsRules	173-174	✓	Policy Group
Forecast Errors	175-177	✗	Policy Group
Water Leases	178-183	✓	Policy Group
Exchanges	184-187	✓	Policy Group
Check Initial and Time Series Values	188-190	✗	Policy Group
Set Misc Data	191-194	✓	Policy Group

Figure 1.2. Screen Capture of the Names for All the Policy Groups in the URGWOM RiverWare Ruleset

## **II. Synopses of Policy Controls**

The key driving forces that affect the policy for operating the facilities in the Rio Grande basin within New Mexico are the following: the trans-basin diversion of water from the San Juan River to the Rio Chama and paper accounting for San Juan water versus native Rio Grande water, flood control operations, the interstate Compact, diversions by the Middle Rio Grande Conservancy District (MRGCD) and other water users including the six Indian pueblos in the Middle Valley, releases to meet downstream target flows to comply with the ESA, and operations for recreational interests. Operations are significantly complicated by the separate accounting for San Juan water versus native Rio Grande water. Note that throughout this document, there are numerous references to Rio Grande water versus San Juan water or Rio Grande releases versus San Juan releases. These terms are used to refer to the two different colors of water in the system. Also, the discussion in this Section of the report often refers the reader to the discussions for the individual rules in Appendix A for more detailed information.

### ***2.1. San Juan-Chama Project***

The San Juan-Chama Project was authorized by Congress in 1962 and allows for the trans-basin diversion of water from the San Juan River to the Rio Grande basin. The project is part of the Colorado River Storage Project to allow for New Mexico to utilize its 11.25% share of the Colorado River water as stipulated in the Upper Colorado River Compact (1948).

Water is diverted based on the capacity restrictions of the diversion facilities while assuring minimum bypass flows in the San Juan tributaries (Refer to Table 2.1). There is a maximum diversion volume allowed in any year and over any 10-year period. The water is conveyed to Azotea Creek in the Rio Grande basin and flows into Heron Reservoir which is on Willow Creek. The fixed firm yield that is allocated to the different contractors for various purposes is 96,200 acre-ft. Refer to Table 2.2 for the allocations for contractors in the Water Operations model and Table 2.3 for allocations in the Planning Model that includes a Combined account for several contractors for simulation efficiency for long-term planning studies.

Table 2.1. Minimum Bypasses at the San Juan Basin Diversions (cfs)

MONTH	RIO BLANCO (Blanco Diversion)	LITTLE NAVAJO RIVER (Little Oso Diversion)	NAVAJO RIVER (Oso Diversion)
January	15	4	30
February	15	4	34
March	20	4	37
April	20	4	37
May	40	27	88
June	20	27	55
July	20	27	55
August	20	27	55
September	20	27	55
October	20	4	37
November	20	4	37
December	15	4	37

Table 2.2. Allocations to Contractors for San Juan-Chama Project Water in the Water Operations Model  
(HeronData.SanJuanContractorAllocations)

Contractor	Allocation (acre-ft)
Albuquerque	48,200
MRGCD	20,900
Jicarilla	6500
Santa Fe	5605
Cochiti Rec Pool	5000
Department of Energy	1200
PVID	1030
Espanola	1000
Belen	500
Bernalillo	400
Taos	400
Los Lunas	400
Red River	60
Twining	15
Uncontracted	2990
San Juan Pueblo	2000
TOTAL ALLOCATION:	96,200

Table 2.3. Allocations to Contractors for San Juan-Chama Project Water in the Planning Model  
(HeronData.SanJuanContractorAllocations)

Contractor	Allocation (acre-ft)
Albuquerque	48,200
MRGCD	20,900
Cochiti Rec Pool	5000
Combined	22,100
<b>TOTAL ALLOCATION:</b>	<b>96,200</b>

The San Juan Diversions at the Blanco, Little Oso, and Oso diversions are computed in URGWOM with consideration for the total annual limit (270,000 acre-ft/year), limited diversion for a decade (1,350,000 acre-ft/10 years), or the available space at Lake Heron.

The annual and 10-year diversion limits are input to the SanJuanChamaRules data object. If the diversion is limited based on one of these criteria, a different algorithm may be used to compute each diversion if the total diversion will be less than 150 cfs. Otherwise and if the diversions are not input, the diversions are set based on the input diversion capacities of 520 cfs for the Blanco diversion, 650 cfs for the Oso Diversion, and 150 cfs for the Little Oso diversion and with consideration for the minimum bypasses presented in Table 2.1. The capacities are input in the Max Diversion table slots for the corresponding diversion objects, and the minimum bypasses are input to the reach objects for the San Juan tributaries. Refer to the discussion for the San Juan Diversions rule for more information on how the San Juan diversions are computed in the model.



Figure 2.1. Photo of Little Oso Diversion  
(courtesy of Reclamation)



Figure 2.2. Photo of the Oso Diversion  
(courtesy of Reclamation)

San Juan water is allocated to contractors on January 1 of each year if water is available. The contracted allocations (Refer to Tables 2.2 and 2.3) are input to the SanJuanContractorAllocations table slot in the HeronData data object. If the storage in the federal pool at Heron Reservoir is greater than the total allocation for all contractors, the full allocation is then made to all the contractors.

Otherwise, the contractors receive a portion of the federal pool based on the ratio of their contracted allocation to the sum of the contracted allocations for all the contractors. Additional allocations up to the contracted total allocation are made in the same manner on July 1 if full allocations could not be made on January 1. Refer to the discussion for the SetSanJuanContractorAllocations and SetSanJuanContractorAllocationsJuly1 rules for more details on how these allocations are made in the model.

Within the model, separate accounting is included to track San Juan-Chama Project water versus native Rio Grande water throughout the system as San Juan water is delivered to contractors or to storage accounts. Storage of San Juan-Chama Project water is only authorized at Heron Reservoir, so native Rio Grande water is required to be bypassed and therefore is not to be stored. The contractors do not have dedicated storage space at Heron Reservoir, so if contractors have not called for their water by December 31, they lose their allocation and it remains in Heron Reservoir as part of the project supply, or federal pool (i.e. there is no carryover of annual allocations). The Bureau of Reclamation (Reclamation) may issue waivers to allow for contractors to store water in Heron Reservoir until as late as September 30<sup>th</sup> of the following year. This has been allowed because of the resulting ecological benefits and operational flexibility. San Juan-Chama Project water will not be released from Heron Reservoir if it is spilling or it has full ice coverage as determined with the HeronSJReleaseRestrictions rule.

Contractors for San Juan-Chama Project water may cause depletions in the basin and then use allocated San Juan-Chama Project water to payback the river. Schedules for these paybacks are input in URGWOM. Actual paybacks are determined by the Office of the State Engineer using the regional groundwater model (depletions are generally caused by groundwater pumping), and the deliveries are requested through letters from the Office of the State Engineer to Reclamation, hence the name “letter water deliveries.” Within URGWOM, the Exchanges Manager in RiverWare is used to establish debts for contractors to deliver water to the location of the Otowi stream gage object in the model based on input delivery schedules. Separate accounts are set up for the delivery of water from allocated storage space for individual contractors to Otowi, and these debts are established using the SetNoAlbuquerqueLoanEXs rule and the SetAlbuquerqueJemezEXs rule.

San Juan-Chama Project water may be released from Heron Dam for letter water deliveries or to fill downstream allocated storage space. Any water released from Heron Reservoir for use by contractors below El Vado Dam is simply passed through El Vado Reservoir. MRGCD may store San Juan water at El Vado Reservoir if the space is available above the storage of native Rio Grande water (MRGCD may allow for other contractors to use this storage space). MRGCD will keep San Juan-Chama Project water in storage and deliver the water to the Middle Valley to meet their diversion demand after all native supplies have been exhausted.

Some contractors have contracts to store San Juan water at Abiquiu Reservoir up to a pool elevation of 6220 ft with the majority of the space allocated to the Albuquerque



Bernalillo County Water Utility Authority (Albuquerque). The actual allocated storage space decreases due to sediment accumulation. Albuquerque will store water until it is needed to meet the demand at their surface water diversion or for letter water deliveries. San Juan water at Cochiti Lake is simply passed through except for the San Juan water delivered to maintain the 1200 acre recreation pool at Cochiti Lake.

Within the model, priorities are input to identify which contractors receive San Juan water first if for some reason all computed demands cannot be met. While there is not actually a priority system, such priorities must be input to the model based on release type and then by account. The priorities are input in the ReleaseTypePriority and AccountReleasePriority periodic slots in the data object corresponding with each reservoir. At Heron Reservoir, priorities for the destination are also considered after release type but before account priorities. The user-defined RuleIsInPriority function is referenced in the ruleset to ultimately reset the priorities for the rules, or the order that certain rules fire, for delivering San Juan water.

## **2.2. Flood Control Operations**

All the reservoirs are operated to comply with flood operations criteria and downstream channel capacities as possible. If the pool elevation at a reservoir exceeds a designated threshold, releases will be adjusted accordingly. If there is an unregulated spill, the releases will be reduced as possible per flood control policy to comply with the downstream channel capacities. Releases will also be adjusted to comply with stepped release restrictions as possible. The computed outflows include consideration for restrictions due to the physical constraints of the outlet works.

At Heron Reservoir, the change in the pool elevation over one day is restricted to be no greater than 1.0 ft (Refer to the discussion for the HeronCheckDeltaStorage rule for more details). At El Vado Reservoir, if the pool elevation exceeds the maximum allowable elevation of 6901.0 ft, the outflow will be set to the release required to reduce the pool elevation down to that maximum allowable elevation or the maximum possible release through the outlet works if that maximum possible outflow is lower (Refer to the discussion for the ElVadoFloodControl rule for more information). As a higher priority, the release will be restricted to be no greater than the downstream channel capacity of 5000 cfs (Refer to the write-up for the ElVadoChannelCapacity Rule in the ElVado policy group).



Figure 2.3. Photo of El Vado Dam (courtesy of Reclamation)

At Abiquiu Reservoir, water will be “pre-evacuated” before April 30 if the forecasted inflow will increase the pool elevations at El Vado Reservoir and/or Abiquiu Reservoir



over either of the corresponding maximum allowable elevations by a certain threshold. The pre-evacuation release is set to the average required release to evacuate the forecasted inflow before August 1. The stepped release restrictions as captured by the AbiquiuSteppedRelease rule and the downstream channel capacities of 1800 cfs below the dam, 3000 cfs at Chamita, and 10,000 cfs at Otowi as checked with the AbiquiuChannelCapacityRestrictions rule are higher priority. As an even higher priority, a maintenance outflow can be input to override normal operations (Such a maintenance flow would be used to simulate a unique situation). If there is an unregulated spill at Abiquiu Reservoir, the releases will be reduced as possible to comply with the downstream channel capacities. The policy for Abiquiu Reservoir includes an additional stipulation that allows for water that was stored under flood operations to be carried over until after the irrigation season starting on July 1 and after the flow at Otowi drops below 1500 cfs. That carry over storage is then released from November 1 through March 31 based on the current inputs, but these dates can be changed.

Flood control operations at Cochiti Dam and Jemez Dam involve consideration for the combination of the outflows from each dam. The releases are adjusted based on the ratio of the available space at each reservoir to the total available space in both reservoirs in order to comply with the channel capacity at Central, currently input as 7000 cfs, and San Marcial, currently input as 5000 cfs in the Planning model (Refer to the CentralChannelCapacityRule, SanMarcialChannelCapacityRule, JemezSanMarcialChannelCapacity, and CochitiChannelCapacityRestrictions rules). The channel capacity at San Marcial is lower due to the current constraint at the railroad bridge. Stepped release restrictions at both Cochiti Dam and Jemez Dam as captured by the CochitiSteppedRelease rule are higher priority. If necessary, releases will be adjusted to balance operations as stipulated in the water control manuals for each facility using the CochitiWCMBalancedRelease rule. If there is an unregulated spill at Cochiti Dam, the release through the gates will be reduced to comply with the downstream channel capacity as possible using the CochitiFloodControl rule.

At Elephant Butte Dam, if necessary, the release will be increased to reduce the pool elevation down to the corresponding winter or summer “prudent” elevation. That release is restricted to be no greater than the downstream channel capacity of 5000 cfs. The prudent pool elevation is based on having 50,000 acre-ft of space available below the top of the conservation pool from March through October (summer) and 25,000 acre-ft of space vacant from November through February (winter) (Refer to the discussion for the ElephantButteOutflowRestrictions rule).

At Caballo Dam, the release will be set to the maximum outflow if the pool elevation exceeds the maximum elevation of 4182.0 ft. Below that maximum elevation, the release will be set to the appropriate value in the flood release table if that lookup value will reduce the pool elevation below the threshold flood pool elevation of 4172.45 ft, or the inflow will be bypassed if that inflow is greater than the outflow required to reduce the pool elevation below that flood pool elevation (Refer to the write-up for the CaballoFloodControlReleaseBasedOnFORD rule). As a lower priority policy, the release

from Caballo Dam will be reduced as possible to comply with the downstream channel capacity of 11,000 cfs at El Paso as coded for the CaballoOutflowRestrictions rule.

### **2.3. Compact**

Equations are coded into expression series slots in the RioGrandeCompact data object (and RioGrandeCompactMonthly data object) in the model for completing New Mexico calculations for the Rio Grande Compact (1938). The calculations involve determining the native Rio Grande flow at Otowi excluding San Juan-Chama Project water. The Otowi Index Supply is computed and the corresponding scheduled delivery to Elephant Butte is determined (Refer to the flowchart in Figure 2.4 for the steps for this calculation with reference to each expression series slot in the model). The actual delivery to Elephant Butte Reservoir is also computed, and a flowchart for the calculation is displayed in Figure 2.5. An annual net adjustment to the Compact credit is calculated as the difference between the Elephant Butte Effective Supply and the scheduled delivery to Elephant Butte. The annual net adjustment is applied at the end of each calendar year to the running Compact credit. Evaporation losses are applied to a Compact credit during the year (but not a Compact debt).

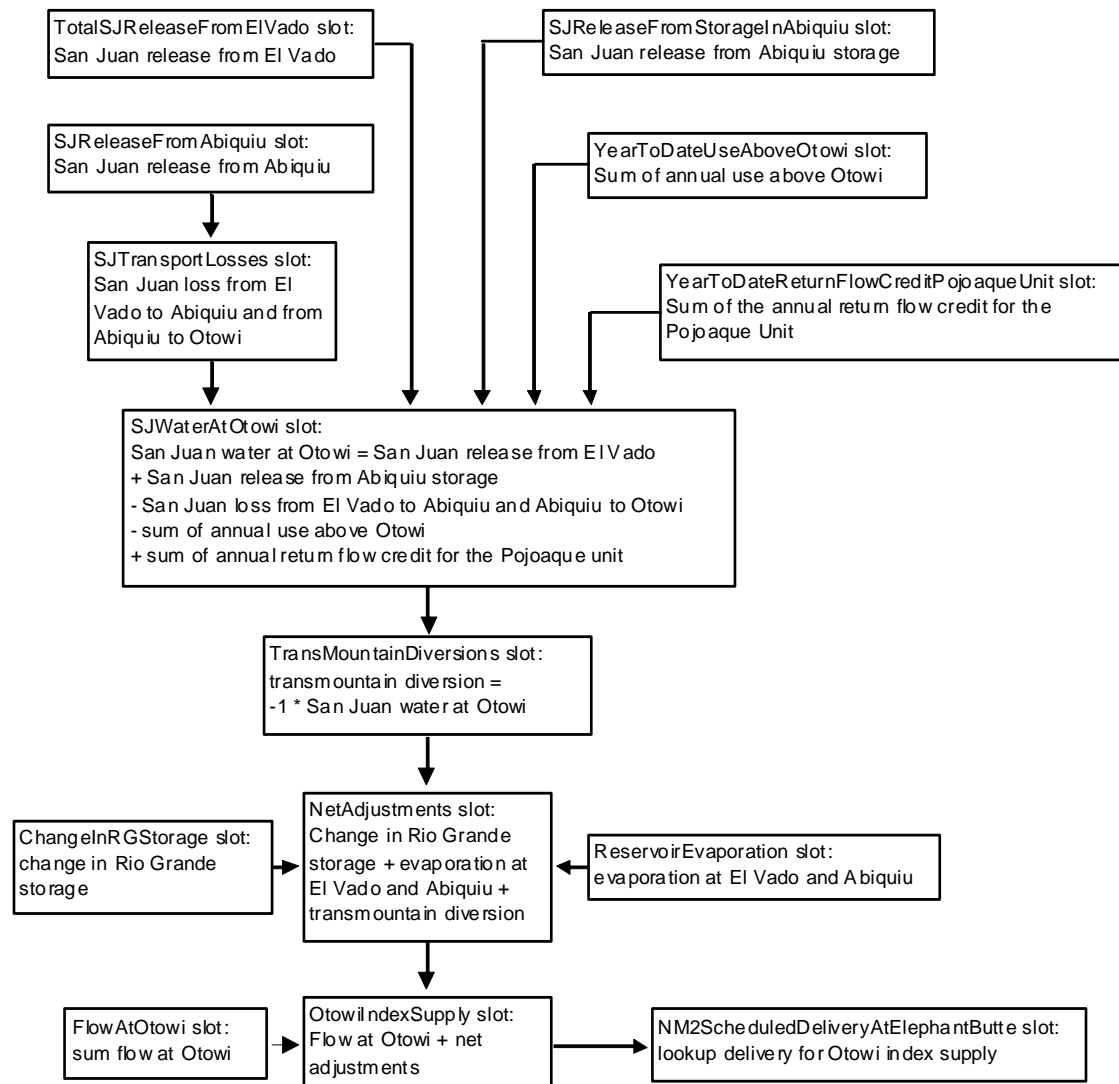


Figure 2.4. Flowchart for New Mexico Compact Calculations – Compute Otowi Index Supply and Scheduled Delivery at Elephant Butte

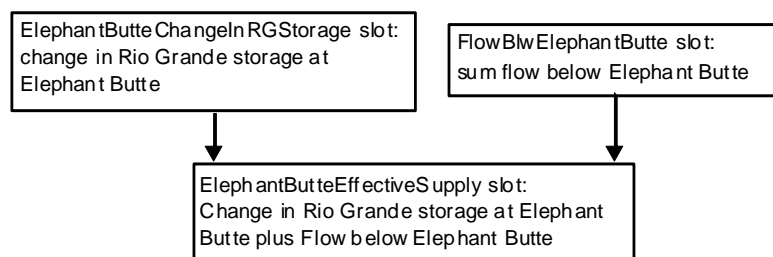


Figure 2.5. Flowchart for New Mexico Compact Calculations – Compute Elephant Butte Effective Supply

### **2.3.1. Article VII of the Compact**

Under Article VII of the Compact, it is stipulated that native Rio Grande water cannot be stored in post-1929 reservoirs when the usable storage is less than 400,000 acre-ft. This does not include the Prior and Paramount storage for the pueblos as discussed in Section 2.4.3. The usable storage is computed with the CompactVIIUsableStorage rule and includes the storage at Elephant Butte Reservoir and Caballo Reservoir minus credit water stored at Elephant Butte Reservoir or any San Juan-Chama Project water as tracked in the accounting. For each timestep in the model, a switch is set with the SetCompactVIISwitch rule to identify whether the stipulations of Article VII of the Compact are in effect for that day. That switch is then referenced throughout the ruleset. Article VII status affects storage of native Rio Grande water at El Vado Reservoir.

### **2.3.2. Relinquished Credits and Emergency Drought Water**

Agreements have been made in the past where Compact credits are relinquished and allocations are made for storage of native Rio Grande water at El Vado Reservoir as Emergency Drought water when stipulations of Article VII of the Compact are in effect. Policy is coded to simulate relinquished Compact credits and the subsequent storage of Emergency Drought water. If this aspect of policy is modeled, the assumption is that Compact credits will be relinquished annually on an input date each year if the Compact credit exceeds an input threshold to reduce the credit to an input target credit after a relinquishment (SetRelinquishedNMCredits rule). The magnitude of a relinquishment can also be input. Allocations for subsequent storage of Emergency Drought water at El Vado Reservoir are set in the UpdateEmergencyDroughtStorageAllocations rule based on input fractions (summing to one) of the relinquished credit for each of three purposes: MRGCD, ESA, and municipalities.

Inflows of native Rio Grande water to El Vado Reservoir when Article VII is in effect are then stored to separate accounts for Emergency Drought water after any storage requirement for Prior and Paramount needs is met first (Refer to Section 2.4.3 for more information on P&P storage). Storage accumulates in the Emergency Drought accounts with the actual inflow of native Rio Grande water. The allocation for storage of Emergency Drought water for municipalities is tracked but is not currently used since exact policy for how such water would be used by municipalities has not been defined. MRGCD water is used to meet their diversion demand, and ESA water is used to meet target flows in the Middle Valley.

If needed, storage of Emergency Drought water could potentially be modeled as reregulation storage at Abiquiu and/or Cochiti Reservoirs using the RioGrandeConservation accounts at these reservoirs as this operation has occurred in the past.

## **2.4. Diversions**

Diversions are set in the model for numerous diversions on the Rio Chama, the four main MRGCD diversions, and the Albuquerque surface water diversion.

### **2.4.1. Rio Chama Diversions/Acequias**

There are five aggregate diversion site objects in the model that represent diversions from the Rio Chama. The Scull and Monastery diversions are included together above Abiquiu Reservoir. The diversion requested for each individual diversion is input to the RioChamaDiversionData object and referenced with the SetRioChamaDiversionAndDepletionRequested rule. The depletion requested is also set with the same rule based on the input fractional return flow in the RioChamaDiversionData object. The input fractional return flow is currently set to 0.5 for all the Rio Chama diversions for all months. Other diversions are located below Abiquiu Dam, and the diversion and depletion requested values for each are set in the same manner. The Chili diversion, the Chamita and Hernandez diversions, and the Salazar diversion are subtracted at three separate locations further downstream. Note that there will be no storage of water at El Vado Reservoir if it would deprive these downstream acequias of water.

### **2.4.2. MRGCD Diversions**

Diversions by MRGCD at Cochiti, Angostura, Isleta, and San Acacia are set with the SetMiddleRioGrandeDiversionRequested rule based on input diversion requested values in the MiddleRioGrandeDiversionData data object. Diversions from Cochiti Lake to the Sili and Eastside Main canals are subtracted in the model immediately below Cochiti Dam. Diversions at Angostura to the Atrisco Feeder and Albuquerque Main Canal are subtracted using the AngosturaDiversions aggregate diversion site object. The input diversions to the Belen High Line Canal and Peralta Main Canal along with the Cacique Acequia, Chical Acequia, and Chical Lateral are subtracted at the Isleta diversion. The requested diversions to the Low Flow Conveyance Channel at the San Acacia diversion dam are also input to the MiddleRioGrandeDiversionData data object. Diversions to the Socorro main canal are set with consideration for flows from the Unit 7 drain.

Releases from upstream storage for the MRGCD diversions are set with reference to a single demand at Cochiti Lake. The demand curve reflects the use of return flows at downstream diversions. Native Rio Grande water in storage at El Vado Reservoir will be released to meet the demand as needed if water is available. Any available Emergency Drought water will be used thereafter, and MRGCD San Juan-Chama Project water will be used after all native supplies have been exhausted.

### **2.4.3. Six Middle Valley Pueblos**

There are six pueblos in the Middle Valley: Cochiti, Santo Domingo, San Felipe, Santa Ana, Sandia, and Isleta. Reservoir operations may entail releasing water if the supply is insufficient to irrigate the 8,847 acres of irrigated land at the pueblos. The Indian irrigation season runs from March 1 to November 15. Irrigated acreage for the pueblos is not distinguished from MRGCD land in URGWOM and diversions to the pueblos are included with MRGCD diversions; however, the storage and release of Prior and Paramount (P&P) water for the pueblos to assure the P&P demand is met is tracked separately in URGWOM.

An initial P&P storage requirement is set on March 1st (or April 1st if not set on March 1st) with the SetElVadoIndianStorageReqAprilAndMarch rule. The storage requirement includes additional storage needed for any dead storage (or unavailable storage below the outlet works) at El Vado Dam. The Indian storage requirement is then updated for each subsequent month for the storage required to meet the demand for the remainder of the irrigation season using the SetElVadoIndianStorageReqAfterApril rule.

The storage requirement is computed as an estimated storage required to meet a monthly demand for the remainder of the year minus the estimated flow that will be available for the remainder of the year as computed with reference to an Otowi forecast volume. A release is made from P&P storage at El Vado if the flow from the mainstem, based on the modeled flow at Embudo, would not meet the demand for the pueblos. This release is made independently of the available supply for MRGCD. The needed release is computed with the IndianStorageRequirementRelease rule.

#### **2.4.3.1. Increased Angostura Diversions**

When MRGCD is in a shortage situation which is indicated when the MRGCD Demand at Cochiti cannot be met with available water in storage for MRGCD, diversions at Angostura are then increased from the regular diversion requested values to the total canal capacity of 400 cfs. This adjustment is completed with the ResetAngosturaDiversionForShortageOps rule. These increased diversions assure water is delivered to the pueblos and reflect adjustments in MRGCD operations during shortage situations such that the limited supply is used most efficiently.

### **2.4.4. Albuquerque Diversion**

The Albuquerque diversion is set up as a water user object in the model, and the diversion requested is computed with the SetAlbuquerqueDiversion rule. The diversions are set to double the full demand for Albuquerque San Juan-Chama Project water as input to the MiddleValleyDemands data object unless it is determined that the diversion should be curtailed or cut off. Releases of Albuquerque's San Juan-Chama Project water are set to provide 65 cfs with loss rates applied. The loss rate is based on the San Juan-Chama loss rate of 1.23 percent from Abiquiu to Cochiti and monthly loss rates from Cochiti to

the diversion. A threshold of 135 cfs for native Rio Grande flow is input for curtailing the diversions per the permit and 70 cfs is input as the threshold for native Rio Grande flow for cutting off the diversions.

Coded policy also includes a check against preemptive cutoff criteria where a preemptive cutoff is implemented before actual permit restrictions would result in curtailed diversions or also during high flows. The preemptive cutoff represents the assumption that Albuquerque would switch to groundwater during low flows before curtailment per the permit would occur or during high flows when it may be unsafe or impractical to operate the diversion dam or when flood control operations at Abiquiu or Cochiti might prevent Albuquerque from receiving a delivery of their allocated San Juan-Chama Project water. The high flow thresholds for a preemptive diversion cutoff are set to 1800 cfs out of Abiquiu or 4500 cfs out of Cochiti. The threshold low flow for a preemptive cutoff is 200 cfs and diversions will not restart until at least two weeks after any preemptive cutoff criterion is not satisfied and the flow at Central is greater than 250 cfs.

Wastewater returns from Albuquerque are set as an input based on historical data and are not affected by a cutoff to the surface water diversions as actual wastewater returns are not dependent on whether surface water or groundwater is being used to provide drinking water. Assumed returns range from approximately 77.5 cfs to 83.4 cfs (slightly more than half the diversion).

## **2.5. Downstream Target Flows**

Releases of supplemental water for targets are made in the model with consideration of the physical losses as represented by all the different methods in the model. The physical model was calibrated with reference to results from a simulation for historical operations versus historical gage data. Supplemental water consists of leased San Juan-Chama Project water or Emergency Drought water stored at El Vado Reservoir that is specifically designated for targets.

Targets are input to a table and established for hydrology year types: dry, average, or wet. Targets in the table may be adjusted based on an input adjustment factor. A step down in targets after the continuous flow requirement is included in the current target table, and additional step downs at Isleta, San Acacia, and San Marcial may be implemented for discretionary operations as discussed further in Section 2.5.3. Targets are identified at each timestep using the `MinCentralTarget` and `MinIsletaSanAcaciaSanMarcialFlowTargets` rules. The Central targets may be modified to provide recruitment or overbank flows as part of Cochiti deviations.

### **2.5.1. Hydrology Year Type**

The year classification for setting targets in the model is set using the Hydrology Year Type rule based on a March through July forecasted flow volume at Otowi calculated

with reference to input inflows. This approach essentially matches the actual approach of referencing the runoff forecast. The year will be classified as dry or wet if the forecasted flow volume is less than 80% of the average Otowi flow volume or greater than 120% of average, respectively. The determined year classification on May 1<sup>st</sup> is maintained for the remainder of a calendar year. A year is classified as dry regardless of the forecast if the stipulations of Article VII of the Compact are in effect, but since the year classification is set for the remainder of the year on May 1<sup>st</sup>, the year classification will not change if the Article VII status changes after May 1<sup>st</sup>.

### **2.5.2. Adjustment Factor**

An adjustment factor is included in URGWOM to increase targets by a percentage (i.e. a target of 100 cfs will increase to 125 cfs with a 25% adjustment factor). A 25% adjustment factor is currently applied to targets because the model can set releases from Abiquiu to hit targets in the Middle Valley with much better precision than can be done in actual operations. Uncertainty about conveyance losses, MRGCD returns, local inflows, etc. combined with the travel time from Abiquiu to target locations and other physical operational constraints prevent actual releases from being adjusted with such precision, so an adjustment factor is applied to targets in the model such that modeled supplemental water releases more accurately reflect actual release volumes.

### **2.5.3. Discretionary Operations**

URGWOM is set up to simulate discretionary operations as part of the Biological Opinion which entail using supplemental water to manage recession after the runoff and control the rate of drying after river rewetting for minnow salvage. Coded policy for representing discretionary operations entails implementing a 30-day step down in targets at the end of the runoff and 7-day step downs in targets thereafter following each river rewetting event. Magnitudes for the initial flow for the step downs in targets are set to 50, 100, and 50 cfs for initial targets at Isleta, San Acacia, and San Marcial with targets decreasing to zero in five steps for the initial 30-day step down and 7 steps for the subsequent 7-day step downs.

### **2.5.4. Supplemental Water Needed For Targets**

Daily needed releases from Abiquiu Dam to meet targets at Central, Isleta, San Acacia, and San Marcial are computed in the model using hypothetical simulations, or separate side simulations in RiverWare used to iterate and solve for the upstream flow needed to meet a target at a downstream location. Needed releases are determined for the downstream targets. The separate side simulations include consideration for all MRGCD diversions and estimated returns and any diversions by Albuquerque. Four instances of hypothetical simulation are completed for each target location with the rules that include ComputeReleaseToMeetMinimum... in the rule name and the highest needed flow at



Cochiti is determined with the SetCochitiMinimumFlow rule. The determined minimum flow is a *total flow* needed for targets that includes water needed for MRGCD and Albuquerque diversions. Another instance of hypothetical simulation is completed for the segment of the model from Abiquiu Dam to Cochiti Lake to determine the total flow needed from Abiquiu Reservoir as determined in the AbiquiuTotalFlowToMeetTarget rule.

The amount of supplemental water needed from Abiquiu is computed by subtracting the release of native Rio Grande water, letter water deliveries, any release of MRGCD San Juan-Chama Project water, and the release of Albuquerque's San Juan-Chama Project water for the surface water diversion from the determined total flow needed at Abiquiu for targets. This calculation is completed in the ComputeAbiquiuMinFlowsDemand rule. The resulting amount of supplemental water needed varies daily based on the estimated physical losses, but the needed release of supplemental water, as set with the SetAbiquiuMinFlowsDemand rule, is not adjusted until the supplemental water needed based on the physical losses has changed by more than 50 cfs and will not be adjusted twice within three days. The computational approach for adjusting the releases is configured such that the volume for the release of supplemental water approximately matches the volume of supplemental water needed based on the hypothetical simulations. Refer to Figure 2.6 for a flow chart that depicts the series of steps for setting a release of supplemental water from Abiquiu Reservoir starting with the targets at the four target locations.

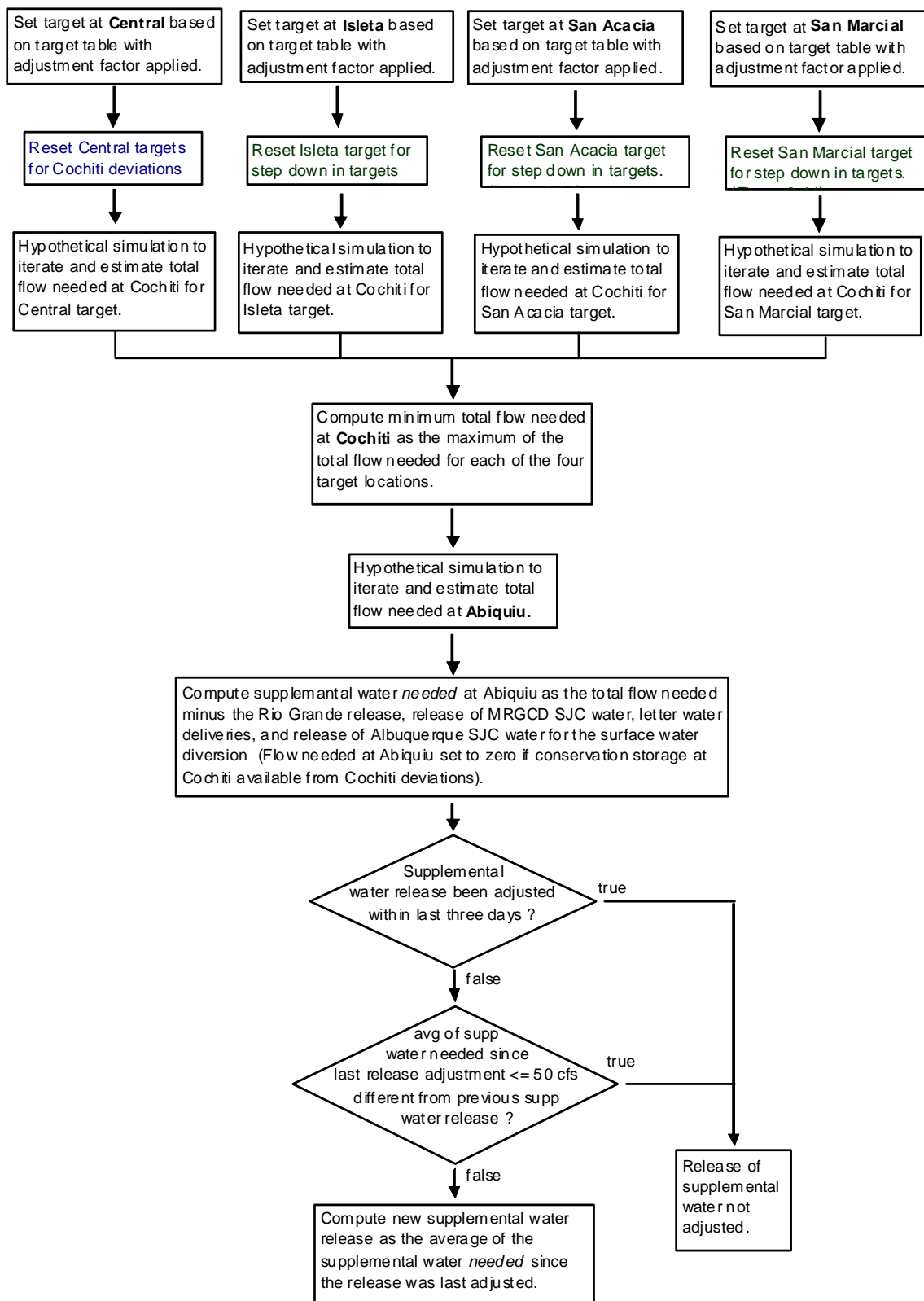


Figure 2.6. Flow Chart for Logic to Set Release of Supplemental Water from Abiquiu

### **2.5.5. Shorted Diversions**

If MRGCD is in a shortage situation and does not have the supply to meet their full demand, it is possible that full MRGCD requested diversions would not be met. Requested diversions may then be shorted in the model in such a shortage situation to prevent the diversion of released supplemental water that is specifically designated for meeting target flows. This adjustment is implemented in a simulation with the Shorted Middle Valley Diversions rule by completing additional side simulations to iterate and solve for what the diversions would be without supplemental water, and requested diversions at the main MRGCD diversions are reset from the full requested diversions to the determined shorted diversion. This assumption is currently only applied if there are downstream targets. If there are no downstream targets, it is assumed that any supplemental water still in the river is available for diversion. For example, if supplemental water is released to meet a target flow at Central, diversions may be shorted at Cochiti or Angostura if MRGCD would not have received their full requested diversion at those locations without supplemental water, but if there were no targets below Central, remaining supplemental water in the river at Isleta may be diverted at the Isleta diversion during a shortage situation.

### **2.5.6. Pumping from the Low Flow Conveyance Channel**

URGWOM is set up to model pumping of flows from the Low Flow Conveyance Channel (LFCC) to the river to prevent river drying. Refer to Figure 2.7 for a picture of pumps used to pump from the LFCC. Diversions at the Neil Cupp site, North Boundary of the Bosque del Apache National Wildlife Refuge, and South Boundary are simulated (Pumping at the Fort Craig site was determined to be inconsequential to URGWOM simulation results and is not included). Water that seeps into the Low Flow Conveyance Channel is pumped to the river where pumping begins based on specific river flow triggers. Different triggers could be established as a function of the year classification for setting targets; although, the threshold low San Acacia flow triggers for initiating pumping at each site are the same in the current model regardless of the year type. Different values can be set up for the winter of wet years.

After pumping has initiated at a site, pumping will continue for a minimum of one week and until the flow at San Acacia has exceeded 150 cfs. Pumping will cease for the year at each site after input dates for each site. It is assumed in the current model that pumps at the Neil Cupp site and North Boundary would not be used after June 30<sup>th</sup>. This aspect of operations can be turned off fairly easily by setting the date to shut off pumping at each site to January 1<sup>st</sup>.



Figure 2.7. Low Flow Conveyance Channel Pumps

## **2.6. Recreation**

There are two key recreational factors that affect policy in the basin. A portion of the San Juan water is used to maintain a permanent recreation pool of 1200 acres at Cochiti Lake, and a rafting release from El Vado Dam may be set as a lower priority operation. The annual allocation of San Juan-Chama Project water for the Cochiti Rec Pool is 5000 acre-ft, and the water is delivered to maintain a content of 49,370 acre-ft plus any sediment accumulation at Cochiti Lake which corresponds to a reservoir surface area of approximately 1200 acres (Refer to the CochitiRecPoolMaxVolume table slot in the HeronData data object). Rules for setting a rafting release from El Vado Dam are turned off in the current ruleset and need to be updated. Actual operations entail moving contractor San Juan-Chama Project, as possible, on weekends starting Memorial Day weekend or by the July 4th holiday through Labor Day weekend to provide rafting flows.

### III. Priorities for Rule Execution

The rules in the URGWOM ruleset execute, or fire, in reverse order from the lowest priority rule to the highest priority rule (Rule 1) and are sorted from upstream to downstream for setting reservoir releases. The initial lowest priority rules in the Set Misc Data and Check Initial and Time Series Values policy groups are used to set initial values for calculations to be completed and check for necessary inputs, respectively. Next, exchanges for loans or leases are identified with rules in the Exchanges and Water Leases policy groups. Exchanges represent a paper record of loans for San Juan-Chama Project water between different contractors. A percent forecast error may be computed, with rules in the Forecast Errors policy group, which is referenced when forecasting river flows at Otowi or inflows to Abiquiu Reservoir.

After such initialization rules are executed, rules in the Diversions and Demands policy group are used to evaluate the current state of the system and identify key parameters that are considered when setting releases. The system storage is evaluated to see if the stipulations of Article VII of the Compact are in effect, the type of year for identifying downstream target flows is evaluated, and inflows are forecasted. Also, the demands throughout the basin are identified, and requested diversions are set.

For each dam, there are essentially three sets of rules. The first series of preparatory rules fire to compute various parameters to then be referenced when the releases of native Rio Grande water and San Juan water are computed. These calculations focus on identifying the downstream demands and evaluate other aspects of operations that drive the need for releases. Initial values for releases of Rio Grande water and San Juan-Chama Project water are used to calculate an initial total outflow from the dam. These rules basically represent the normal range of operations.

A second series of rules is then used for each reservoir to determine a final total outflow from the dam with consideration for physical and legal constraints where outflows can then be altered if not within the normal range of operations, either due to high flows (flood control operations) or low flows (minimum releases). Flood control operations tend to have the highest priority (i.e. lower rule number) because of safety and damage considerations. The outflow is set with consideration for flood control operation criteria if reservoir pool elevations exceed specific thresholds. The outflow will be restricted to comply with downstream channel capacities as possible, and stepped release restrictions may also be a priority. The computed total outflow will also be checked to assure it exceeds a designated minimum outflow. The determined final outflow is always checked to assure the release is legitimate based on the outlet works for the dam. If there is an unregulated spill, the total release is adjusted as much as possible by using the regulated outflow through the gates at the dam. Given the reservoir pool elevation, the outflow may also be restricted based on the rating curve for the outlet works.

After the total outflow has been set, a final reconciled outflow of Rio Grande and San Juan water is computed and used to identify the actual values releases from individual

accounts. A third series of rules is used for each dam to update accounting slots that identify the delivery of San Juan water (or Rio Grande water) to payback debts, fill storage accounts, transfer water between different accounts, or simply be passed through the reservoir toward a downstream destination. Specific details about policy at each dam are presented below.

### 3.1. Heron

Releases of Rio Grande and San Juan-Chama Project water from Heron Dam under normal operations are computed with the `ComputeHeronRGRelease` and `EstimateHeronSJRelease` Rules, respectively, and a predetermined total outflow is set to the sum with the `HeronOutflow` Rule. Rio Grande water is effectively bypassed, but operations actually entail making releases during the runoff and evacuating Rio Grande water during other periods after storage accumulates. Within URGWOM, the calculation entails computing a different Rio Grande release depending on whether it is the first half of the month or second half.



Figure 3.1. Photo of Heron Reservoir

San Juan-Chama Project water is released from Heron Reservoir to fill downstream allocated storage space for contractors or to meet other contractor demands. If the ice coverage at Heron Reservoir is equal to a value of 1.0 (i.e. 100% ice coverage) as input to the `HeronData` data object, the release of San Juan-Chama Project water is reset to zero. The ice coverage is included as part of the `CurrentSurfaceAreaPanAndIce` evaporation method within `RiverWare`. Also, if the elevation at Heron Dam is greater than the Maximum elevation of 7186.1 ft and Heron Dam must spill, the release of San Juan water is reset to zero. Refer to the discussion for the `HeronSJReleaseRestrictions` Rule for



Figure 3.2. Photo of Heron Dam  
(courtesy of Reclamation)

more details on coding for these two criteria. Flood operation criteria are checked with the `HeronCheckDeltaStorage` Rule, and the final outflow of Rio Grande water is reconciled with the `SetHeronRGAccount` Rule. If the total release exceeds the predetermined release of Rio Grande water, the Rio Grande outflow is reconciled to equal that lower total outflow. If the total release exceeds the sum of the predetermined release of Rio Grande and San Juan water, the Rio Grande outflow is reconciled to include that additional outflow.

### **3.2. El Vado**

For El Vado Dam, the predetermined release of Rio Grande and San Juan water is computed with the ComputeElVadoRGRelease and ComputeElVadoSJRelease Rules, respectively, and a predetermined total outflow is set to the sum of the two with the ElVadoOutflow Rule with consideration for a computed minimum release. MRGCD San Juan-Chama Project water is stored at El Vado Reservoir as space is available with the storage of native Rio Grande water. Native Rio Grande water is stored, as *not* needed to meet downstream demands, if the stipulations of Article VII of the Compact are not in effect. Otherwise, all native Rio Grande water is bypassed if not needed for P&P storage or Emergency Drought storage. Flood operations and the downstream channel capacities are checked with the ElVadoFloodControl and ElVadoChannelCapacity Rules, respectively, and the final outflow of Rio Grande water is reconciled with the SetElVadoRGAccounts Rule.



Figure 3.3. Photo of the Outlet Works at El Vado Dam

### **3.3. Abiquiu**

Predetermined releases of Rio Grande and San Juan water from Abiquiu Dam are computed with the ComputeAbiquiuRGRelease and ComputeAbiquiuSJRelease Rules, respectively. The subsequent total outflow is set as the sum of the two. Rio Grande water is bypassed at Abiquiu Dam unless storage is required for flood control operations or is being modeled as proposed conservation storage. Releases of San Juan-Chama Project water include bypassed inflows and releases from storage for contractors to meet



downstream demands including the Albuquerque surface water diversions. The predetermined total outflow is checked against a 25 cfs minimum outflow.

Any water stored for flood control operations may be locked in as carryover storage between July and October if 212,000 acre-ft of space is available at Cochiti and the flow at Otowi drops below 1500 cfs before the stored water can be evacuated. That carryover storage is then released from November through March (or other periods for the release of carryover storage can be defined). If it is determined that water needs to be pre-evacuated at Abiquiu Reservoir in anticipation of large forecasted inflows, the outflow from Abiquiu Dam is reset based on a computed pre-evacuation flow which includes consideration for the downstream channel capacity and stepped release restrictions (Refer to the AbiquiuPreEvacuation Rule). The stepped release and downstream channel capacity restrictions are checked with the AbiquiuSteppedRelease and AbiquiuChannelCapacityRestrictions Rules, respectively. For Abiquiu Dam, a maintenance flow may be input. Flood operations criteria are checked with the AbiquiuFloodControl Rule, and the final outflow of Rio Grande water is reconciled with the SetAbiquiuRGAccount Rule.



Figure 3.4. Photo of the Outlet Works at Abiquiu Dam

### **3.4. Cochiti and Jemez**

Releases from Cochiti and Jemez Dams are essentially computed together. The predetermined releases of Rio Grande and San Juan water for both dams are computed with the rules in the Preparatory Cochiti And Jemez policy group. The total outflow from



Cochiti Dam is set to the predetermined release of San Juan water plus the input minimum Rio Grande outflow of 25 cfs while assuring the Cochiti Rec pool is maintained and the demands in the Middle Valley for MRGCD are met and also considering whether storage should be locked in at Cochiti Lake until after the irrigation season (Refer to the discussion for the Cochiti Outflow Rule). All San Juan-Chama Project water is bypassed except for Cochiti rec pool water, and native Rio Grande water is also bypassed unless storage is needed for flood control operations.

The minimum release to meet downstream target flows is checked with CochitiTargetFlowOutflow Rule. The channel capacities at Central and San Marcial are checked with the CochitiChannelCapacityRestrictions Rule with specific consideration for the combination of the flows from both Cochiti and Jemez Dams. Stepped release restrictions at Cochiti Dam are checked with CochitiSteppedRelease Rule, and the rule for balanced operations of Cochiti and Jemez Dams as identified in the water control manuals for each dam are checked with the CochitiWCMBalancedRelease Rule. Flood operations criteria are checked with the CochitiFloodControl Rule, and the final outflow of Rio Grande water is reconciled with the SetCochitiRGAccount Rule. Inflows to Jemez Reservoir are bypassed (The model is set up to be able to simulate a sediment pool, which is not currently used).

### **3.4.1. Cochiti Deviations**

Cochiti deviations are authorized through 2013 where the U.S. Army Corps of Engineers (Corps) may temporarily store native Rio Grande water to be released at the time of the peak and augment flows to provide recruitment flows in the Middle Valley (Corps, 2009). Specific criteria are coded for identifying whether the runoff is sufficient to enact Cochiti deviations to provide recruitment flows (or overbank flows) for the benefit of species listed under the Endangered Species Act but the runoff is insufficient for providing the needed hydrograph by just bypassing inflows at Cochiti Lake. Operations entail providing overbank flows if conditions support providing the higher flows. This aspect of policy can be set to expire based on an input year as the last year that Cochiti deviations are authorized – currently set to 2013.

Deviations will be implemented to provide recruitment flows if the March through July Otowi flow forecast is between 50% and 80% of average and the projected peak inflow to Cochiti Lake during the recruitment or overbank season is between 1800 and 5000 cfs or the March through July forecast is greater than 80% of average but the projected peak inflow is less than 3500 cfs. The projected peak inflow to Cochiti is estimated during an URGWOM simulation based on input inflows. Deviations will be implemented to provide overbank flows if the Otowi forecast is between 80% and 120% of average and the projected peak inflow to Cochiti is between 3500 and 10,000 cfs or the Otowi forecast is between 50% and 80% of average but the projected peak inflow is greater than 5000 cfs.

The date to start storage at Cochiti Lake for deviations can be input to URGWOM, but if no preset date is input in the current model, the date to begin storage is set to 24 days before the projected date of the peak inflow to Cochiti Lake. Target flows to provide recruitment or overbank flows are input as 30-day target hydrographs. If deviations are implemented, targets at Central are reset such that day five in the appropriate target hydrograph matches the date of the projected peak inflow to Cochiti Lake (Refer to the WriteNewMinTargetsForRecruitmentHydrographKnowledge and WriteNewMinTargetsForOverbankHydrographKnowledge rules).

Water in reregulation storage for Cochiti deviations is released as needed for targets where the needed release reflects the adjusted targets at Central to provide either recruitment or overbank flows. Remaining water in reregulation storage will then be evacuated by the end of a deviations period which lasts for 45 days as currently input (ComputeCochitiRGConservationRelease rule). Water will begin to be evacuated 15 days before the end of the Cochiti deviations period at a constant rate if that constant rate is greater than the flow needed to meet targets. When the reregulation storage drops below 3000 acre-ft, targets are adjusted back to the original Central targets with the EndTargetsForOverbankOrRecruitment rule. This adjustment is needed to prevent other sources for supplemental water (i.e. leased San Juan-Chama Project water or Emergency Drought water) from being used to meet the recruitment or overbank targets.

### ***3.5. Elephant Butte and Caballo***

Releases from Elephant Butte and Caballo Dams are initially set with ElephantButteOutflow and CaballoOutflow Rules based on an input full demand schedule and a computed percentage where the percentage reflects the portion of the full demand that can be provided based on the water in storage and the forecasted runoff. Downstream channel capacities are checked along with criteria for maintaining flood storage space.

## IV. Other Topics

Descriptions of RiverWare and rulebased simulation are not provided in this report. New users of RiverWare or URGWOM are encouraged to attend the Simulation, Rulebased Simulation, and Accounting training courses held by CADSWES. The online help documentation should also be reviewed. There are a few particular aspects of RiverWare that are specifically discussed here to prepare new users for working with the URGWOM ruleset.

### ***4.1. Planning Model versus Water Operations Model***

There are four daily timestep modules of URGWOM: the forecast, accounting, planning, and water operations models. The forecast model and the accounting model are not discussed in this report; however, the planning model must be distinguished from the water operations model when working with the ruleset. The planning model is used to complete long-term simulations for comparative analyses and planning studies; whereas, the water operations model is used to simulate operations for the coming year for preparing annual operating plans (AOP) for forecasting operations, river flows, reservoir levels, etc.

The primary difference between the Planning Model and the Water Operations Model is the number of accounts. For the Water Operations Model, sixteen contractors for San Juan-Chama Project water are represented; whereas, for the Planning Model, several contractors are combined into one Combined account resulting in just four contractors, and this affects the rules in a few locations, but the same URGWOM ruleset is used with both models.

A data object named WaterOpsOrPlanning is included with each model file that contains a scalar slot named Trigger. The Trigger slot value is set to 1 in the Water Operations Model and 0 in the Planning Model. A function called IsWaterOpsRunNotPlanningRun in the General Boolean Functions utility group is referenced at different locations in the ruleset to see if the ruleset is being used with the Water Operations Model or the Planning Model. Rules specific for either model can be referenced or skipped by checking the IsWaterOpsRunNotPlanningRun function in the execution constraints for the rule. Differences in the code for some select rules due to the different number of accounts are also isolated with references to this function. For example, when allocations are made at Heron Reservoir for the contractors, the equations are different depending on whether the ruleset is being used with the Water Operations Model or with the Planning Model. The appropriate equation is accessed by referencing the IsWaterOpsRunNotPlanningRun function.

## **4.2. User-defined vs. Predefined Functions**

Within RiverWare, there are numerous predefined functions. Several of these predefined functions are used in the URGWOM ruleset, but this document does not include descriptions of these predefined functions in RiverWare. Discussions of these functions are presented in the RiverWare online help. Logic for many of the user-defined functions is discussed in this document within the rule logic write-ups for each rule. Most of the functions used in the URGWOM ruleset are user-defined functions that were specifically created using the RiverWare rule policy language (RPL) by the URGWOM developers. The user-defined functions are grouped in separate utility groups. Within the discussion of the logic for the individual rules, the name for the utility group where a particular function is located is identified.

## **4.3. Accounting**

The accounting functionality within RiverWare is not extensively documented but is a major component of URGWOM. Separate accounts are setup to track the use and movement of San Juan water versus native Rio Grande water. The status of individual accounts is key for setting reservoir releases and identifying how and which demands will be met. There are 4 different identifiers for the accounts: Water Type (San Juan vs. Rio Grande), Account Type (Storage, Pass-through, or Diversion), Release Type, and Destination. Release types are identified primarily such that assignments can be made for multiple accounting supplies in one rule with a single release type such as “flowthrough.” The name for a destination includes the contractor and destination. For example, the Albuquerque storage account at Abiquiu Reservoir would be given a destination name of *Albuquerque\_Abiquiu*.

### **4.3.1. Supplies**

The primary role of the rules for the accounting is to set supplies to transfer water each timestep. A transfer may be from one account to another account to payback a debt or to deliver water from one location to another location. Supplies are also set when San Juan water will simply be passed through a reservoir to be delivered to a contractor or a storage account. Note that there are many supplies for pass-through accounts. The nomenclature for supply names are based on the following format: *ContractorName-Origin-ContractorName-Destination*. For example, the supply with a name of *MRGCDAbiquiuToReclamationAbiquiu* is the name for the supply to transfer water from the MRGCD account at Abiquiu Reservoir to Reclamation’s account at Abiquiu Reservoir.

Note that within the rules, the names for supplies are seldom directly input. The names are usually identified with one of several user-defined functions in the Account Lists utility group such as the *AllSuppliesBySourcesReleaseTypeDestinations* function. These functions may even be used to identify the name for one supply. New users of the

URGWOM ruleset are encouraged to specifically review these functions before working with the rules. For some cases, the logic in these functions is discussed with the rule logic for the individual rules but often, the rule logic discussions focus on the more unique aspects of each rule.

### **4.3.2. Exchanges**

An exchange manager is used in RiverWare to track loans for San Juan water from one contractor to another. The exchanges are established to maintain a record of the debt, since that debt may not be paid back at the same timestep that the debt is established. The exchange then essentially represents a follow-up demand on a source for San Juan water such as a reservoir. The exchange manager is a separate tracking system for monitoring the loans and the demand for a payback. The nomenclature for exchange names is based on the following format: *ContractorName-Destination-ContractorName-SourceLocation-EX*. Each timestep, supplies are set to payback the debts based on input priorities for the use of San Juan water and the availability of San Juan water stored in and released from the source storage account.

The primary use of exchanges in URGWOM is to establish debts for contractors' letter water deliveries to payback the river at Otowi based on the input schedules in the DeliveryRequests data object. These debts are set each timestep using either the SetAlbuquerqueLoanEXs or SetNoAlbuquerqueLoanEXs rules in the Exchanges policy group. Debts are also set with the SetHeronElVadoEXs and SetAlbuquerqueJemezEXs rules in the same policy group. Exchanges are also used for the past practice where Albuquerque would loan water to other contractors since Albuquerque historically did not need their full allocation (Refer to the Rules in the Exchanges policy group). Loans are paid back with different rules in the Heron San Juan Accounts, ElVado San Juan Accounts, and Abiquiu Accounts policy groups that set supplies with reference to the user-defined GetSortedPaybackAccounts function.

## **4.4. Other RiverWare Features**

Within the rules, slots are often checked to see if a value is a NaN, which stands for Not-a-Number. Slots will be populated with NaNs until values are input or assignments are made during the rulebased simulation. If a value is a NaN, no assignment has been made yet for the identified timestep. If the value is not a NaN, an assignment has already been made, and this is often checked to prevent assignments from being attempted again. This is done such that the simulation will be completed more efficiently. It is also important to understand that a simulation begins at the start timestep, but the timestep before the start timestep is the initial timestep which may contain many key input values that are needed at the start timestep.

Several subbasins are identified in URGWOM. Subbasins are automatically created in RiverWare but may also be established as user-defined subbasins. The subbasins are lists

of objects in the model that can then be referenced by the rules. Multiple assignments can be made with one rule for identical slots in all the objects within a subbasin. The subbasins are also referenced for the hypothetical simulations to identify the portion of the system to be modeled as part of the hypothetical simulation. The predefined HypTargetSimWithStatus function is referenced several times in the URGWOM ruleset to complete hypothetical simulations as side calculations with all the objects in an identified subbasin. Refer to the RiverWare documentation for more information on subbasins. The subbasins in URGWOM can be viewed by clicking on the Edit Subbasins label in the Workspace menu.

List management is used extensively in the URGWOM ruleset where a function may be used to create a two or three-dimensional array with a list of slots to be set with a rule, a list of dates to be set, and/or a list of values to be set. Within the rule, values in the created list of lists are referenced using the GET STRING @INDEX, GET DATETIME@INDEX and GET NUMERIC @INDEX statements. Multiple assignments for setting numerous diversions or accounting supplies can be set within one rule using this approach.

## **V. Conclusions**

Documentation of the RiverWare rules in the URGWOM ruleset version 4.1.2 was prepared to serve as a reference for URGWOM users. This report provides summaries of key aspects of the policy coded in URGWOM including the impact of San Juan-Chama Project water on operations in the basin, stipulations for flood control operations, Compact calculations, diversions to acequias, MRGCD, Albuquerque, and the six pueblos, releases to meet downstream target flows, and recreational interests. The priorities at each dam along the Rio Chama and Rio Grande are summarized. Aspects of RiverWare that are used abundantly in the ruleset and that are key to understanding the policy as coded in URGWOM are discussed. Discussions and flowcharts of the logic used for each rule in the ruleset are also included in Appendix A to serve as a reference for URGWOM users. This report could also be used to complete an initial review of the policy as coded in URGWOM.

## **REFERENCES**

- Rio Grande Compact. 1938. States of New Mexico, Colorado, and Texas.
- Upper Colorado River Compact. 1948. States of New Mexico, Colorado, Utah, Arizona, and Wyoming.
- URGWOM RiverWare Ruleset version 4.1.2. 2010. URGWOM Technical Team
- U.S. Army Corps of Engineers (Corps). 2009. Final Environmental Assessment and Finding of No Significant Impact for a Temporary Deviation in the Operation of Cochiti Lake and Jemez Canyon Dam, Sandoval County, New Mexico.
- U.S. Fish and Wildlife Service (Service). 2003. Biological and Conference Opinions on the Effects of Actions Associated With the Programmatic Biological Assessment of Bureau of Reclamation's Water and River Maintenance Operations, Army Corps of Engineers' Flood Control Operation, and Non-Federal Actions on the Middle Rio Grande, New Mexico. Albuquerque, New Mexico.

## A. Individual Rules

All of the individual rules in the URGWOM ruleset are discussed separately and are grouped based on policy group. There are 26 separate Sections in this appendix for the 26 different policy groups. The rules fire in reverse order within URGWOM, so the discussion of the rules begins with the last rule in each policy group and the last policy group in the ruleset. The discussion for each rule typically includes an Explanation, notes on the Execution Constraints, and a write-up on the Rule Logic. As noted in the discussions, a few rules are specifically included for the Planning Model or specifically for the Water Operations Model. Screen captures of the RiverWare rule policy language are included for a few cases. The level of detail of the discussion and in the flowcharts varies depending on the complexity of the rule. For the more complex rules, some of the finer details of the logic will still need to be referenced in the electronic ruleset file.

### A.1. Set Misc. Data

This policy group includes some basic preparatory rules to assure required data are entered as needed to complete a simulation. The SetCombinedDeliveryRequest and SetAlbuquerqueDeliveryRequest rules are used to set alternate schedules for delivery of letter water to payback the river at Otowi if conditions in the rule are satisfied. The alternate delivery schedules are proposed actions that were analyzed as part of a study for the Collaborative Program using the Planning Model.

#### A.1.1. Set Supply NaNs To Zero

*Explanation:* At the beginning of each simulation, this rule sets initial values for three different accounting supplies to zero if the values are NaNs. Initial values for the release of native Rio Grande water from the separate Indian Storage, Supplemental ESA, and MRGCD Drought accounts at El Vado Reservoir are needed for subsequent calculations and set to zero. For the Supplemental ESA and MRGCD Drought accounts, values for the supplies are set to zero for every day of the calendar year up through the initialization timestep.

*Rule Execution:* The rule executes at the start timestep.

*Rule Logic:* Values for the accounting supplies for the release of native Rio Grande water from the IndianStorage, SupplementalESA, and MRGCD Drought accounts are set to zero if the values are NaN as checked with IF THEN statements. FOREACH loops are used to assign zero values for all the slots from January 1 through the initialization timestep for the releases from the Supplemental ESA and MRGCD Drought accounts at El Vado. Refer to Figure A.1 for a screen capture of the rule policy language from the RiverWare ruleset.



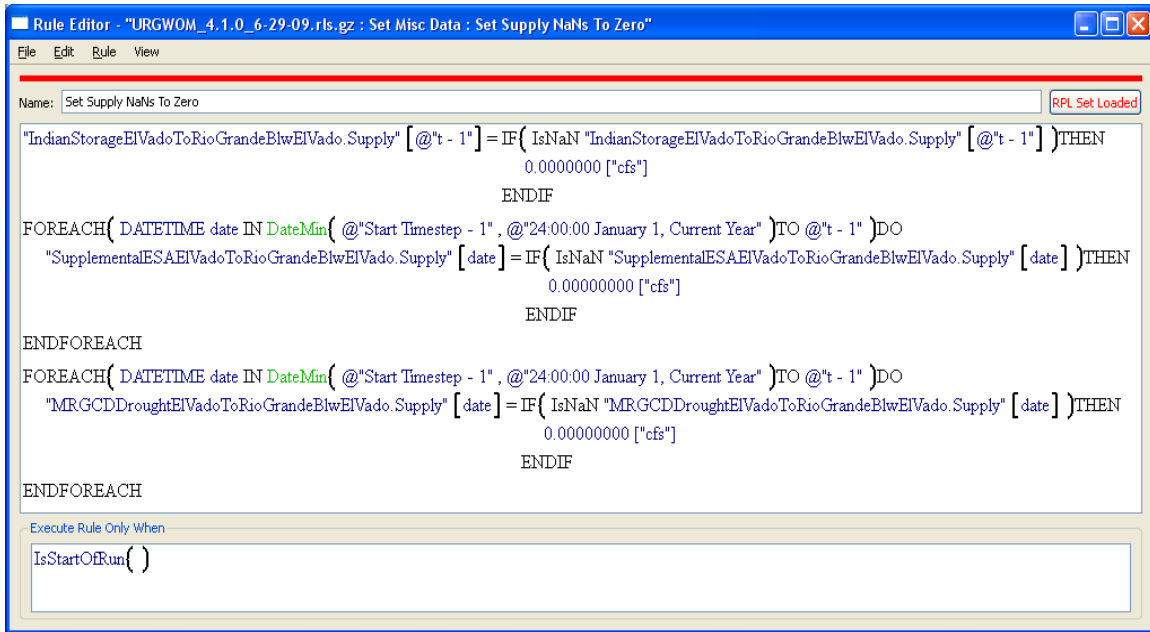


Figure A.1. Rule Policy Language for the Set Supply NaNs To Zero Rule

### A.1.2. Set Average Nambe Falls Data for RGCompact

*Explanation:* At the start of the simulation, average values are set to the UseAboveOtwi and ReturnFlowCreditPojoaqueUnit slots in the RioGrandeCompact data object if values are not input. Values for these two series slots are needed for Compact calculations completed with expression series slots in that same data object. The average values are in the UseAboveOtwiPeriodic and the ReturnFlowCreditPojoaqueUnitPeriodic periodic slots in the RioGrandeCompact data object.

If the Otowi gage flow is input for January 1 through the initialization timestep, assignments are made for each timestep from January 1 through the finish timestep. If the Otowi gage flow is not input from January 1 up to the initialization timestep, average values are only set for the start timestep through the finish timestep. This check for inputs to the Otowi stream gage object for January 1 through the initialization timestep is used to assess whether values have been input to complete Compact calculations for the current year in which case the UseAboveOtwiPeriodic and ReturnFlowCreditPojoaqueUnit values would also be needed back to January 1.

Warning messages are displayed in the diagnostic output window during the run if average values are used. A separate warning message is displayed if Otowi gage data were not input back to January 1 and the start timestep is after January 1.

*Rule Execution:* The rule executes at the simulation start timestep if there are any NaNs entered in the UseAboveOtwi and ReturnFlowCreditPojoaqueUnit time series slots in the RioGrandeCompact data object.

*Rule Logic:* The assignments to the UseAboveOtwi and ReturnFlowCreditPojoaqueUnit slots are completed with FOREACH loops. An IF THEN ELSE statement is used to have the assignments made for January 1 of the current year through the finish timestep if values are input to the Otowi stream gage object for January 1 through the initialization timestep. Otherwise, the assignments are made from the start timestep through the finish timestep. Print statements are used to note that average values were used and if Otowi gage data were not input for January 1 through the start timestep as needed to complete Compact calculations for the current year. Refer to Figure A.2 for a screen capture of the rule policy language from the RiverWare ruleset.

```

Rule Editor - "URGWOM_4.1.0_6-29-09.rls.gz : Set Misc Data : Set Average Nambe Falls Data for RGCompact"
File Edit Rule View

Name: Set Average Nambe Falls Data for RGCompact RPL Set Loaded

FOREACH DATETIME date IN IF { NOT NaNInSlot( Otowi Gage Inflow , @24:00:00 January 1, Current Year , @Start Timestep - 1" ) } THEN DO
    @24:00:00 January 1, Current Year" TO @Finish Timestep"
ELSE
    @t" TO @Finish Timestep"
ENDIF
    RioGrandeCompact.UseAboveOtwi [ date ] = RioGrandeCompact.UseAboveOtwiPeriodic [ date , "Flow" ]
ENDFOREACH
FOREACH DATETIME date IN IF { NOT NaNInSlot( Otowi Gage Inflow , @24:00:00 January 1, Current Year , @Start Timestep - 1" ) } THEN DO
    @24:00:00 January 1, Current Year" TO @Finish Timestep"
ELSE
    @t" TO @Finish Timestep"
ENDIF
    RioGrandeCompact.ReturnFlowCreditPojoaqueUnit [ date ] = RioGrandeCompact.ReturnFlowCreditPojoaqueUnitPeriodic [ date , "Flow" ]
ENDFOREACH
PRINT IF { NaNInSlot( RioGrandeCompact.UseAboveOtwi , @Start Timestep" , @Finish Timestep" ) } THEN
    UseAverageMessage( RioGrandeCompact.UseAboveOtwi )
ENDIF
PRINT IF { NaNInSlot( RioGrandeCompact.ReturnFlowCreditPojoaqueUnit , @Start Timestep" , @Finish Timestep" ) } THEN
    UseAverageMessage( RioGrandeCompact.ReturnFlowCreditPojoaqueUnit )
ENDIF
PRINT IF { NaNInSlot( Otowi Gage Inflow , @24:00:00 January 1, Current Year , @Start Timestep - 1" ) } THEN
    "Not all data is available to sum back to beginning of year for Rio Grande Compact numbers"
ENDIF

Execute Rule Only When
IsStartOfRun( ) AND { NaNInSlot( RioGrandeCompact.UseAboveOtwi , @Start Timestep" , @Finish Timestep" ) AND NaNInSlot( RioGrandeCompact.ReturnFlowCreditPojoaqueUnit , @Start Timestep" , @Finish Timestep" ) }

```

Figure A.2. Rule Policy Language for the Set Average Nambe Falls Data for RGCompact Rule

### A.1.3. SetCombinedDeliveryRequest

*Explanation:* As a potential flow tool for providing additional water for meeting target flows in the Middle Valley, this rule was set up to simulate a proposed action of using an alternate schedule for letter water deliveries for the Combined account if certain criteria are satisfied (similar to the SetAlbuquerqueDeliveryRequest rule). The rule essentially sets the delivery of the portion of letter water designated to payback the Compact at alternate times to evaluate benefits on augmenting flows needed for recruitment or to

help manage recession after the runoff. This rule was developed for a specific flow tool and is not very flexible and would need to be reviewed if the needs for modeling alternate delivery schedules changes. Note that this rule does not fire if values are already input for the Combined series slot in the DeliveryRequests data object. Also, the Combined account is specific to the Planning Model, so this rule would never be used with the Water Operations Model.

Letter water deliveries for Santa Fe and half of the amount for other contractors not including PVID are delivered at an alternate time if the Compact credit is greater than the input value (70,000 acre-ft). That portion will be delivered in a 7-day spike around the peak (Refer to Figure A.3) if Cochiti deviations are not implemented or as a constant release from June 15th through June 30th to help manage recession if the Compact credit is greater than 70,000 acre-ft but Cochiti deviations were implemented. The second alternative presented in Figure A.4 is input to the CombinedScenario2 series slot in the DeliveryRequests data object with a sample No Action delivery schedule (Figure A.5) input to the CombinedScenario1 series slot.

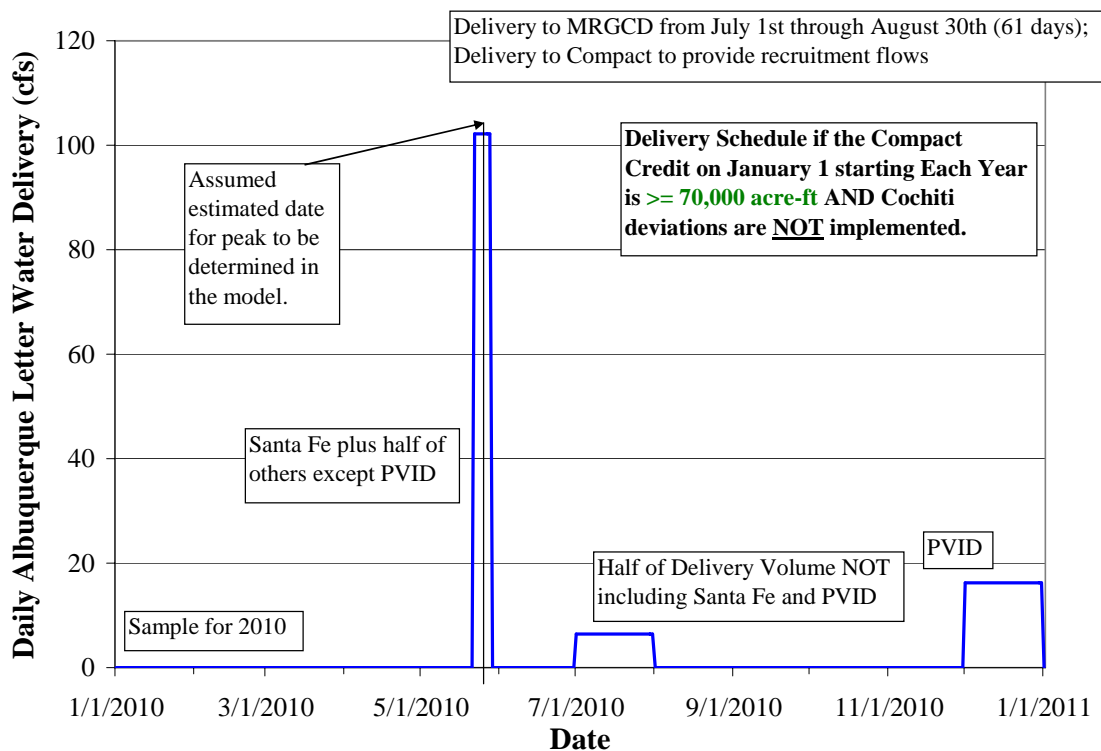


Figure A.3. Sample Alternate Schedule for Combined Account Letter Water Deliveries to Provide Spiked Release

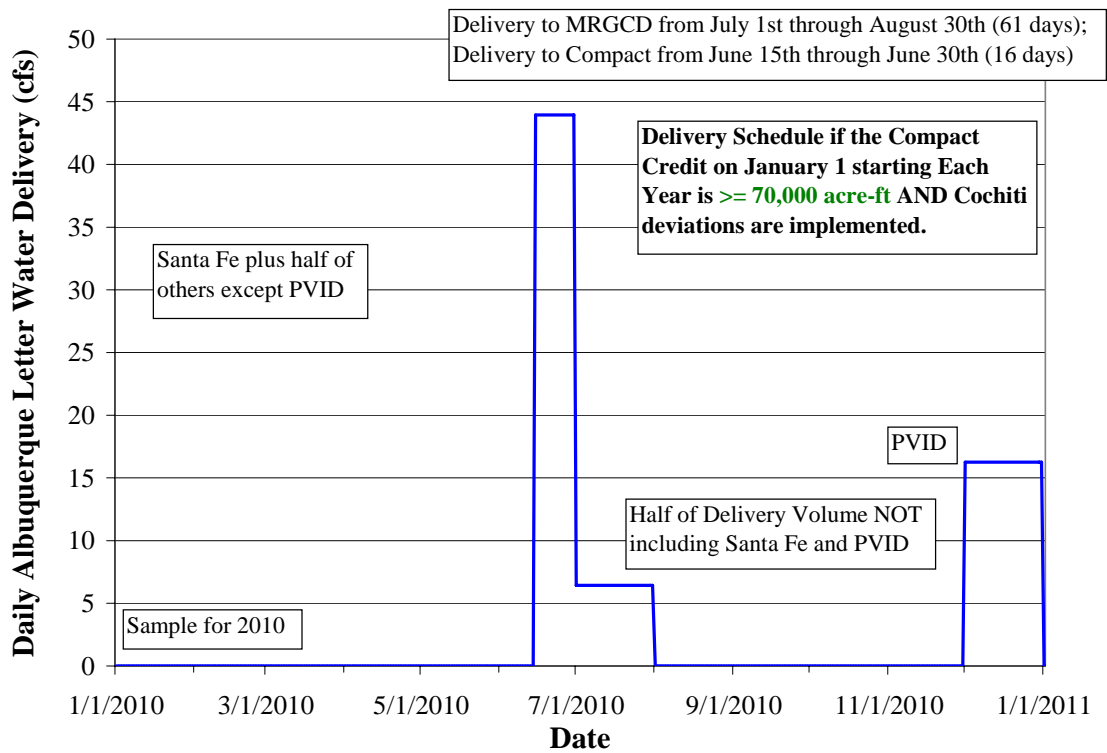


Figure A.4. Sample Schedule for CombinedScenario2 Delivery

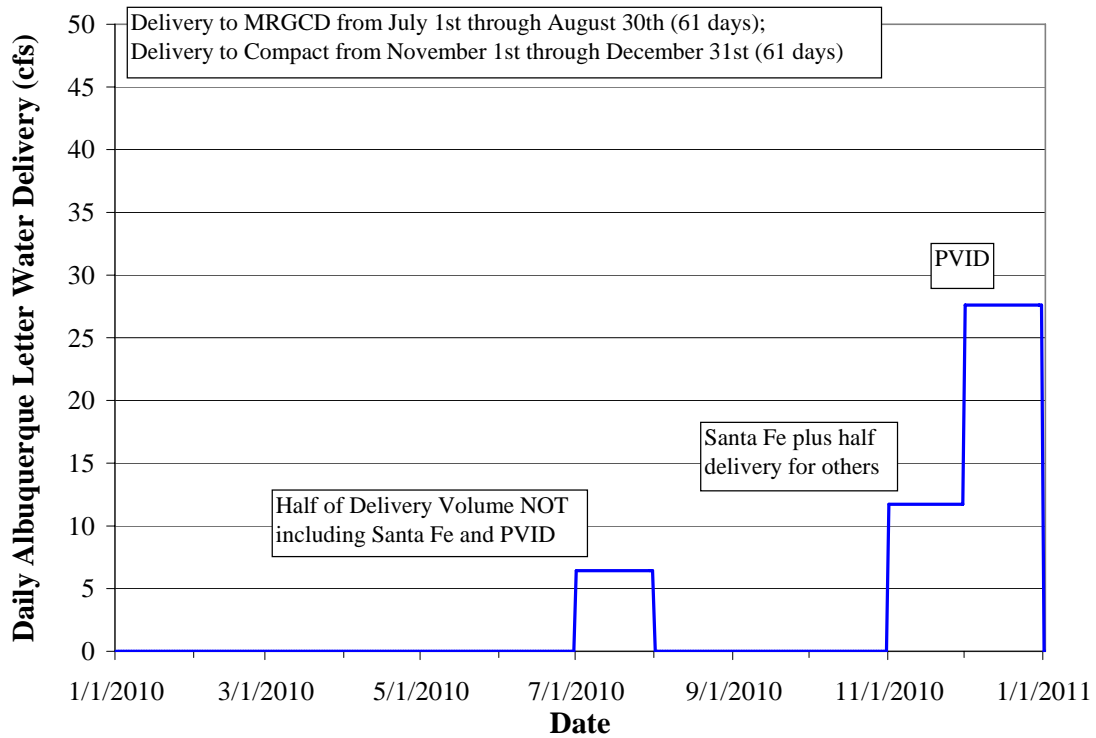


Figure A.5. Sample Schedule for CombinedScenario1 Delivery

*Rule Execution:* The rule executes each timestep if a value is not input directly to the Combined series slot in the DeliveryRequests data object.

*Rule Logic:* IF THEN ELSE statements are used to check the different criteria for alternate delivery schedules.

If the current timestep is at least 3 days before the start of the recruitment or overbank season, the delivery is set to the input value in the CombinedScenario1 series slot in the DeliveryRequests data object for the current timestep (i.e. the baseline delivery schedule is used each year up to three days before the recruitment or overbank season).

Elseif the current timestep is within the period from three days before the recruitment or overbank season and three days after the recruitment or overbank season and the Compact credit exceeds the input value in the CompactCreditForSwitchToScenario2 scalar slot in the DeliveryRequests data object and the Otowi forecast is less than the threshold for implementing Cochiti deviations as checked with the IsNotCochitiDeviationsRange function in the General Boolean Functions utility group (i.e. Cochiti deviations are not implemented to provide recruitment or overbank flows) and the current timestep is within seven days of the peak as checked with the IsWithin7DaysAroundPeak function in the General Boolean Functions utility group, then the delivery is set to the sum of the alternate scheduled delivery for June 15<sup>th</sup> through June 30<sup>th</sup> in the CombinedScenario2 series slot in the DeliveryRequests data object divided by seven (i.e. the portion of delivery input for an alternate delivery schedule for scenario 2 is adjusted to all be delivered in a 7-day spike around the estimated time of the peak to potentially provide recruitment flows).

Elseif the current timestep is greater than three days before the start of the recruitment or overbank season and before June 30<sup>th</sup> and the Compact credit exceeds the input value in the CompactCreditForSwitchToScenario2 scalar slot in the DeliveryRequests data object and the Otowi forecast is less than the threshold for implementing Cochiti deviations as checked with the IsNotCochitiDeviationsRange function in the General Boolean Functions utility group (i.e. Cochiti deviations are not implemented to provide recruitment or overbank flows), the daily delivery is set to zero (i.e. if the criteria were satisfied to provide a 7-day spike, the delivery is set to zero around the provided 7-day spike up to June 30<sup>th</sup>).

Elseif the current timestep is greater than three days before the start of the recruitment or overbank season and the Compact credit exceeds the input value in the CompactCreditForSwitchToScenario2 scalar slot in the DeliveryRequests data object, the daily delivery is set to the value in the CombinedScenario2 series slot in the DeliveryRequests data object (i.e. the previous criteria for providing a 7-day spike were not satisfied, but the threshold Compact credit exceeds the threshold for using the alternate delivery schedule).

Else, the daily delivery is set to the value in the CombinedScenario1 series slot in the DeliveryRequests data object (i.e. the baseline delivery schedule is used if all the previously checked criteria for an alternate delivery schedule were not satisfied).

#### **A.1.4. SetAlbuquerqueDeliveryRequest**

*Explanation:* As a potential flow tool for providing additional water for meeting target flows in the Middle Valley, this rule was set up to simulate a proposed action of using an alternate schedule for letter water deliveries for the Albuquerque account if certain criteria are satisfied (similar to the SetCombinedDeliveryRequest rule). The rule essentially sets the delivery of the portion of letter water designated to payback the Compact at alternate times to evaluate benefits on augmenting flows needed for recruitment or for targets and to prevent river drying. This rule was developed for a specific flow tool and is not very flexible and would need to be reviewed if the needs for modeling alternate delivery schedules changes. Note that this rule does not fire if values are already input for the Albuquerque series slot in the DeliveryRequests data object.

Letter water deliveries from Albuquerque to payback the Compact would be used to provide a 7-day spiked release at the timing of the peak (Figure A.6) if Cochiti deviations are not implemented and the Compact credit is greater than an input threshold (70,000 acre-ft) to the CompactCreditForSwitchToScenario2 scalar slot in the Delivery Requests data object. As a second but lower priority alternate schedule, Albuquerque letter water deliveries to payback the Compact would occur during September and October as opposed to November and December if the same Compact credit threshold is exceeded and the flow at San Acacia is greater than an input threshold (150 cfs) for the last seven days of August as input to the SevenDaySAFlowForAlbScenario2 scalar slot in that data object. The second alternate delivery schedule is input to the AlbuquerqueScenario2 series slot (Figure A.7), and the original No Action delivery schedule (Figure A.8) is input to the AlbuquerqueScenario1 series slot.

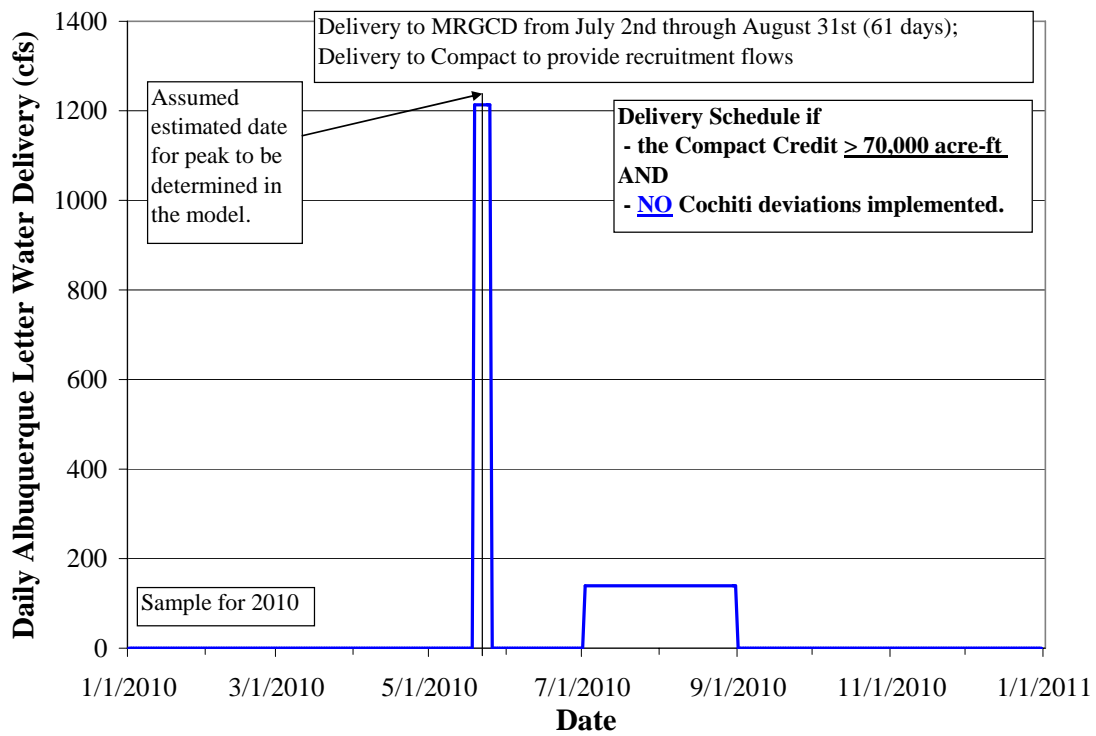


Figure A.6. Sample Alternate Schedule for Albuquerque Account Letter Water Deliveries to Provide Spiked Release

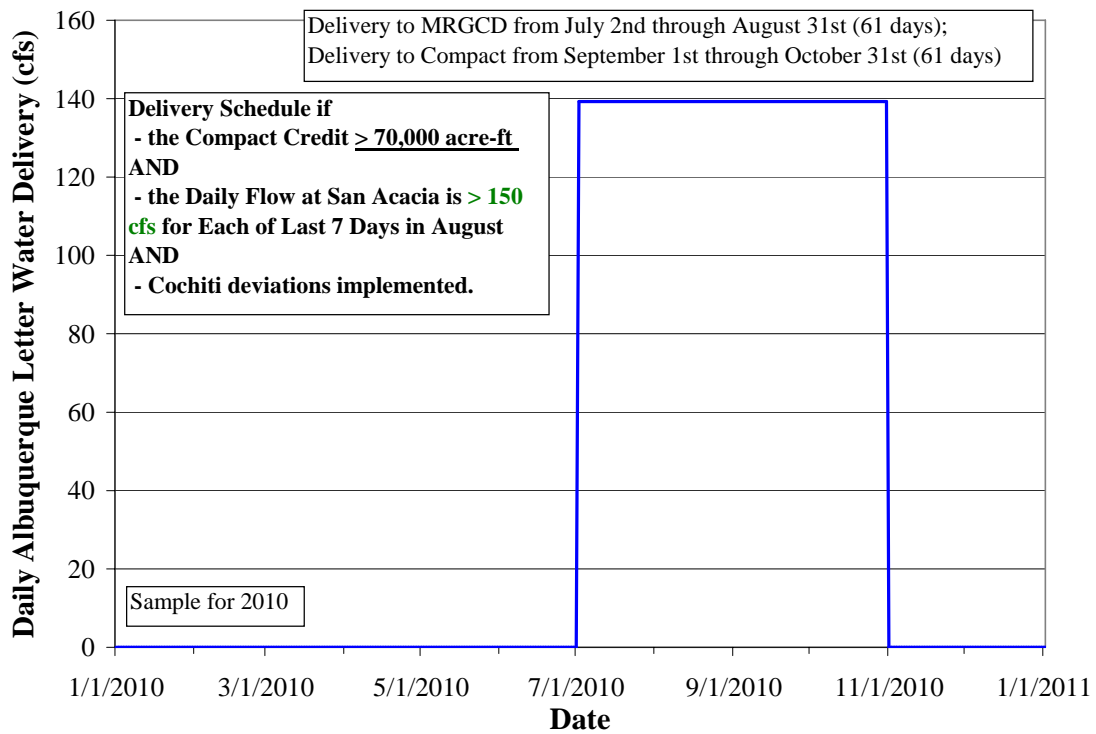


Figure A.7. Sample Schedule for AlbuquerqueScenario2 Delivery

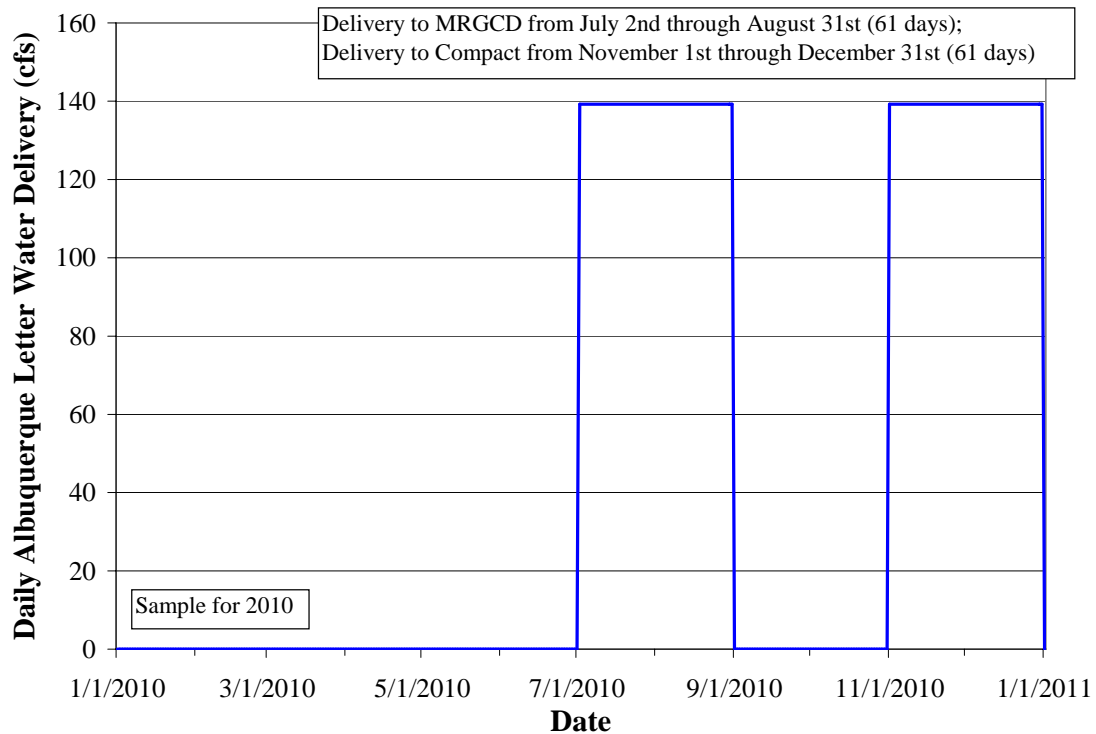


Figure A.8. Sample Schedule for AlbuquerqueScenario1 Delivery

*Rule Execution:* The rule executes each timestep if a value is not input directly to the Albuquerque series slot in the DeliveryRequests data object.

*Rule Logic:* IF THEN ELSE statements are used to check the different criteria for alternate delivery schedules.

If the current timestep is at least 3 days before the start of the recruitment or overbank season, the delivery is set to the input value in the AlbuquerqueScenario1 series slot in the DeliveryRequests data object for the current timestep (i.e. the baseline delivery schedule is used each year up to three days before the recruitment or overbank season).

Elseif the current timestep is within the period from three days before the recruitment or overbank season and three days after the recruitment or overbank season and the Otowi forecast is less than the threshold for implementing Cochiti deviations as checked with the IsNotCochitiDeviationsRange function in the General Boolean Functions utility group (i.e. Cochiti deviations are not implemented to provide recruitment or overbank flows) and the current timestep is within seven days of the peak as checked with the IsWithin7DaysAroundPeak function in the General Boolean Functions utility group and the Compact credit exceeds the input value in the CompactCreditForSwitchToScenario2 scalar slot in the DeliveryRequests data object, then the delivery is set to the sum of the scheduled delivery for the Compact (after September 1<sup>st</sup>) in the AlbuquerqueScenario1



series slot in the DeliveryRequests data object divided by seven (i.e. the portion of delivery input to payback the Compact is adjusted to all be delivered in a 7-day spike around the estimated time of the peak to potentially provide recruitment flows).

Elseif the current timestep is greater than three days before the start of the recruitment or overbank season and before August 31<sup>st</sup>, the delivery is set to the value in the AlbuquerqueScenario1 series slot in the DeliveryRequests data object (deliveries to payback MRGCD are set regardless of the criteria for alternate delivery schedules for the portion to payback the Compact).

Elseif the current timestep is greater than or equal to September 1<sup>st</sup> and the flow at San Acacia exceeded an input threshold for the last seven days of August as checked with the WasSAFlowLastSevenDaysAugEnoughForAlbScenario2 function in the Demand Functions utility group and the Compact credit exceeds the input value in the CompactCreditForSwitchToScenario2 scalar slot in the DeliveryRequests data object and the Otowi forecast is greater than or equal to the threshold for implementing Cochiti deviations as checked with NOT IsNotCochitiDeviationsRange function in the General Boolean Functions utility group (i.e. Cochiti deviations are implemented to provide recruitment or overbank flows), the daily delivery is set to the value in the AlbuquerqueScenario2 series slot in the DeliveryRequests data object (i.e. the previous criteria for providing a 7-day spike were not satisfied, but the threshold Compact credit exceeds the threshold for using the alternate delivery schedule).

Elseif the current timestep is greater than or equal to September 1<sup>st</sup> and the Otowi forecast is less than the threshold for implementing Cochiti deviations as checked with the IsNotCochitiDeviationsRange function in the General Boolean Functions utility group (i.e. Cochiti deviations are implemented to provide recruitment or overbank flows) and the Compact credit exceeds the input value in the CompactCreditForSwitchToScenario2 scalar slot in the DeliveryRequests data object, the daily delivery is set to zero (i.e. if the previous criteria for providing a 7-day spike were satisfied; thus, the delivery for the rest of the year starting on September 1<sup>st</sup> is set to zero because the delivery to payback the Compact was already delivered at an alternate schedule).

Else, the daily delivery is set to the value in the AlbuquerqueScenario1 series slot in the DeliveryRequests data object (i.e. the baseline delivery schedule is used if all the previously checked criteria for an alternate delivery schedule were not satisfied).

## ***A.2. Check Initial and Time Series Values***

The rules in this policy group are used to check for required input values in several slots. These rules take a few minutes to fully execute at the beginning of a simulation if the rules are turned on. After a check of the required inputs has been completed, the rules can be turned off for subsequent simulations. Note that in the current ruleset, the referenced lists of slots with required inputs need to be updated for the current model;

thus, these rules are turned off and have not been used with the current model. An update to these rules has been scheduled.

### **A.2.1. Check Model Inputs Temperature**

*Explanation:* This rule is used to assure input data are entered for the Max Air Temperature and Min Air Temperature in the Cochiti, Jemez, Heron, El Vado, and Abiquiu reservoir data objects.

*Rule Execution:* This rule executes at the start timestep.

*Rule Logic:* A FOREACH loop is used to set the value in the CheckModelInputs series slot on the RuleHasFired data object to 1.0 to indicate that the rule has fired and needed inputs have been set. The assignment resets again to 1.0 after each series is checked. The model run aborts if required inputs have not been made. A list is set up using the reservoir4List function in the Check Input Lists utility group which includes the Max Air Temperature and Min Air Temperature time series slots in the Cochiti, Jemez, Heron, El Vado, and Abiquiu reservoir data objects. For each time series, a FOR WITH DO loop is used to assure there are no NaNs for each time series. The user-defined function NaNInSlotGivenDates function in the General Boolean Functions utility group is used to determine whether NaNs are present for each date of each year of simulation for November 1 through December 31.

### **A.2.2. Check Model Inputs Pan Evaporation**

*Explanation:* This rule is used to assure input data are entered for the Pan Evaporation in the Cochiti, Jemez, Heron, El Vado, and Abiquiu reservoir data objects.

*Rule Execution:* This rule executes at the start timestep.

*Rule Logic:* A FOREACH loop is used to reset the value in the CheckModelInputs series slot on the RuleHasFired data object to 1.0 to indicate that the rule has fired and needed inputs have been set. The assignment resets again to 1.0 after each series is checked. The model run aborts if required inputs have not been made. A list is set up using the reservoir3List function in the Check Input Lists utility group which includes the Pan Evaporation time series slots in the Cochiti, Jemez, Heron, El Vado, and Abiquiu reservoir data objects. For each time series, a FOR WITH DO loop is used to assure there are no NaNs in the time series. The user-defined function NaNInSlotGivenDates function in the General Boolean Functions utility group is used to determine whether NaNs are present for each date of each year of simulation for April 1 through October 31.

### **A.2.3. CheckModelInputsWithListFunctions**

*Explanation:* This rule is used to assure input data are entered for numerous slots throughout the model. All of the slots are included in several different lists in the functions within the Check Input Lists utility group. Note that the otherDataList, initialValueInputList, and waterLeaseList functions used to create lists of slots include a check against the user defined IsWaterOpsNotPlanning function to assure the list of slots appropriately corresponds to the module that is being used with the ruleset.

*Rule Execution:* This rule executes at the start timestep.

*Rule Logic:* A FOREACH loop is used to set the value in the CheckModelInputs series slot on the RuleHasFired data object to 1.0 to indicate that the rule has fired and needed inputs have been set. The model run aborts with a printed message to the diagnostics output window if required inputs have not been set. Lists for different sets of slots are set up using several user-defined functions in the Check Input Lists utility group. WITH FOR DO loops are used to check the inputs for the slots in each list, and IF THEN ELSE statements are used to check the inputs for the appropriate periods depending on the slot.

### **A.3. Exchanges**

Rules in the Exchanges policy group are used to establish contractor debts. These exchanges include the debts for contractors to payback the river (i.e. letter water deliveries for contractors for San Juan-Chama Project water). Provisions are included to allow contractors to borrow water from Albuquerque to payback debts if needed and Albuquerque has water available to loan, determined based on Albuquerque storage that exceeds an input threshold, but this aspect of the rules has not actually been utilized in the latest model applications. Debts are established as exchanges in RiverWare where each exchange has a name, borrow amount, payback destination, and payback source.

#### **A.3.1. SetAlbuquerqueJemezEXs**

*Explanation:* Values for the exchanges for Albuquerque to payback debts to the river and for NMISC to payback the river for impacts from a Jemez sediment pool are set based on input delivery schedules.

*Rule Execution:* If the values for either the RGOtowiAlbuquerqueAbiquiuEX.Borrow or NMISCJemezNMISCAbiquiuEX.Borrow exchanges are a NaN for the current timestep, the rule fires.

*Rule Logic:* The values for the exchanges for Albuquerque to payback the river at Otowi (RGOtowiAlbuquerqueEX.Borrow) and for NMISC to payback the river for the impacts of the Jemez sediment pool (NMISCJemezNMISCAbiquiuEX.Borrow) are set based on the volume of the input delivery request for the current timestep in the DeliveryRequests data object. The predefined VolumeToFlow function is used to convert the volumetric delivery request to a flow before setting the exchange amount.

#### **A.3.2. SetAlbuquerqueLoanEXs**

*Explanation:* Under the condition when Albuquerque has water available to loan, this rule sets exchanges for contractors to payback the river at Otowi *for* the contractors with accounts set up in URGWOM to payback the river from Abiquiu. The debt is established based on the input delivery schedule for that contractor. Note that the actual loans from Albuquerque to the contractors are made with the SetAlbuquerqueLoanAccounts rule. (Refer to the discussion for the SetNoAlbuquerqueLoanEXs Rule which makes assignments if Albuquerque does not have water to loan.)

*Rule Execution:* If this rule has not fired as checked against the predefined HasRuleFiredSuccessfully function and as checked with the AlbuquerqueWillLoan function, the rule fires (i.e. if the rule has not fired and Albuquerque has water to loan, the rule fires). Albuquerque will loan water if the amount of San Juan-Chama Project

water that Albuquerque has in its storage account at Abiquiu Reservoir is greater than the input value for the minimum storage before Albuquerque will loan water (with consideration for a tolerance factor).

*Rule Logic:* The AlbuquerqueAbiquiuEXAccounts function in the Account Lists utility group is used to create a list of accounts in the model for contractors to payback the river at Otowi from water in storage at Abiquiu Reservoir. A list of exchanges is then created for those accounts to establish the debts for contractors to payback the river at Otowi based on the input delivery schedules. The names for the exchanges are identified using the AlbuquerqueLoanEXs function in the Exchange functions utility group which references the MakeOtowiExchangeList function in the Account Lists utility group. Values are set to the input values as determined with the DeliveryRequestFor function in the Get Data Functions utility group. Using a FOREACH loop, the values for all the exchanges are set for the current timestep.

### **A.3.3. SetNoAlbuquerqueLoanEXs**

*Explanation:* This rule sets the values for exchanges for contractors for San Juan-Chama Project water to payback the river for depletions. The exchanges are set for contractors that have accounts in URGWOM to payback a debt to the river at Otowi from Abiquiu, El Vado, and Heron Reservoir. Note that the rule is specifically coded for deliveries to come from specific locations if contractors have multiple accounts in URGWOM to make the payback from different locations. (These priorities are not based on input reservoir priorities utilized elsewhere in the ruleset.) Deliveries are not made from Abiquiu if the contractor also has an account in the model to make the payback from Heron unless the contractor is Albuquerque (The exchange for Albuquerque is set separately with the SetAlbuquerqueJemezEXs rule). Deliveries are not made from El Vado Reservoir if the contractor also has an account in the model to make the payback from Abiquiu. Debts are established for all contractors that have accounts to make the payback from Heron. The Tech Team has planned to edit this configuration.

*Rule Execution:* If this rule has not fired as checked against the predefined HasRuleFiredSuccessfully function and if NOT AlbuquerqueWillLoan is satisfied, the rule fires (i.e. if the rule has not fired and Albuquerque does NOT have water to loan, the rule fires). Albuquerque will NOT loan water if the amount of San Juan-Chama Project water that Albuquerque has in its storage account at Abiquiu Reservoir is less than or equal to the input value for the minimum storage before Albuquerque will loan water (with consideration for a tolerance factor).

*Rule Logic:* Three FOREACH loops are used to set all the exchanges for the three different source reservoirs for the paybacks. Exchanges are first set for the accounts in the model for contractors to payback the river from Abiquiu that do not also have an account to payback the debt from Heron Reservoir (and the contractor is not Albuquerque). The list of accounts is identified with the user-defined AbiquiuEXStorageAccounts function, and the exchange list is determined with the user-

defined MakeOtwiExchangeList function in the Account Lists utility group. Exchanges are then set for the accounts in the model for contractors to payback the river from El Vado Reservoir that do not also have an account to payback the debt from Abiquiu Reservoir. The list of accounts is identified with the user-defined ElVadoEXStorageAccounts function. Exchanges are finally set for all contractors that have accounts in URGWOM to payback the river from Heron Reservoir. Assignments are set equal to the volume of the delivery request, converted to flow, for each account for the current timestep as entered in the DeliveryRequests data object. The DeliveryRequestFor function in the Get Data Functions utility group is used for all three FOREACH loops to identify the input delivery request.

#### **A.3.4. SetHeronElVadoEXs**

*Explanation:* This rule sets the debts for contractors to move water from Heron Reservoir to El Vado Reservoir to assure that contractors have enough water in El Vado Reservoir to meet demands. Additional debts are established above any established debt at previous timesteps for the exchange. No exchange is currently set for Albuquerque. For MRGCD, the debt is established such that deliveries are made, when MRGCD has water available at Heron Reservoir, to assure a minimum storage is maintained at El Vado Reservoir and also assure MRGCD has storage at El Vado needed to make any payback to Albuquerque for a past loan. For other contractors, additional debt is established as needed for contractors to move water to El Vado Reservoir to assure the contractor has storage at El Vado Reservoir for any payback to Albuquerque at Abiquiu Reservoir for a past loan and to make any payback to the river at Otowi from El Vado Reservoir. This rule serves as an additional means for establishing the movement of contractor water to downstream allocated storage space versus other rules that move water to simply fill allocated storage space.

*Rule Execution:* If the value for the MRGCDElVadoMRGCDHeronEX.Borrow exchange is a NaN for the current timestep, the rule fires. This check against one of the exchanges (that exists in both the Planning and Water Operations Model) to be set ensures that the rule only fires once per timestep.

*Rule Logic:* Within the rule, a FOREACH loop is used to set the values for the exchanges. The user-defined HeronElVadoEXs function in the Exchange Functions utility group is used to create a list of the exchange names and values to be set.

In the HeronElVadoEXs function, the MakeHeronElVadoExchangeList function in the Account Lists utility group is used in a WITH DO loop to identify the names for the exchanges.

For the Albuquerque account, the exchange is set to zero. For the MRGCD account, if the storage at El Vado Reservoir is less than the minimum storage and the date is past the target fill date, the exchange is set for MRGCD to deliver water to El Vado Reservoir with water from storage at Heron Reservoir as needed to assure the minimum storage at

El Vado is maintained or as needed by MRGCD at El Vado Reservoir to payback debts to Albuquerque at Abiquiu from El Vado Reservoir for a past loan from Albuquerque. For other accounts, the exchange is set for contractors to move water from Heron Reservoir to El Vado Reservoir as needed at El Vado to payback Albuquerque at Abiquiu for past loans and to also payback the river at Otowi. For other accounts, the exchange for water in storage at Heron is set to payback debts to El Vado Reservoir as available or as needed in El Vado Reservoir to payback debts to Abiquiu and debts to the river at Otowi.

## **A.4. Water Leases**

This policy group includes rules for setting lease amounts by Reclamation from other contractors for San Juan-Chama Project water as transfers at Heron, El Vado, or Abiquiu Reservoirs. Rules are also included to set exchanges for Reclamation and Albuquerque to payback MRGCD for loans and subsequently set supplies for Reclamation to payback the debts to MRGCD at either El Vado or Abiquiu Reservoir and Albuquerque to payback the debt at Abiquiu. Rules are also included to assure input delivery requests as input for contractors can be met with the available supply for each contractor (for the first year of a simulation).

### **A.4.1. Heron Lease Amounts**

*Explanation:* This rule sets supplies to transfer water to Reclamation for leases from contractor accounts at Heron Reservoir based on input schedules. Note that the transfers occur based on the input schedule only if water is available for the source account.

*Rule Execution:* If the value for the AlbuquerqueHeronToReclamationHeron.Supply is a NaN for the current timestep, the rule fires. The check against this single supply set with the rule ensures that the rule only fires once per timestep.

*Rule Logic:* A FOREACH loop is used in the rule to set the supplies for the lease amounts where the supply names and values to be set are established in a list with the ReclamationLeaseList function in the Accounts List utility group. The predefined VolumeToFlow function is used in the rule to convert the volumetric lease amounts to flows.

The list of supply names is created using the user-defined AllSuppliesIntraBySourcesReleaseTypeDestination function with reference to all storage accounts at the reservoir, except CochitiRecPool and NambeFalls accounts would not be included. The lease amounts from each contractor are entered in time series slots in the HeronData data object with the slots named based on the name of the contractor followed by “WaterLease.” The values are referenced using the user-defined AccountLeaseAmount function in the Get Data utility group. The lease amount is checked against the amount of San Juan-Chama Project water that the contractor has

available at the reservoir. Refer to Figure A.9 for a flowchart that depicts the logic used for this rule with specific reference to the accounts in the Water Operations Model.

#### **A.4.2. El Vado Lease Amounts**

*Explanation:* This rule sets the supplies for Reclamation's leases from contractor accounts at El Vado Reservoir. Assignments are made with this rule in the same manner used in the Heron Lease Amounts rule, but the lease amounts are input in the ElVadoData data object. Refer to Section A.4.1 for the discussion of the Heron Lease Amounts rule. Refer to Figures A.9 and A.10 for a flowchart that show the logic for this rule with specific reference to the accounts in the Water Operations Model.

#### **A.4.3. Abiquiu Lease Amounts**

*Explanation:* This rule sets the supplies for Reclamation's leases from contractor accounts at Abiquiu Reservoir. Assignments are made with this rule in the same manner used in the Heron Lease Amounts rule, but the lease amounts are input in the AbiquiuData data object. Refer to Section A.4.1 for the discussion of the Heron Lease Amounts rule. Refer to Figure A.10 for a flowchart that displays the logic used for this rule with specific reference to the accounts in the Water Operations Model.



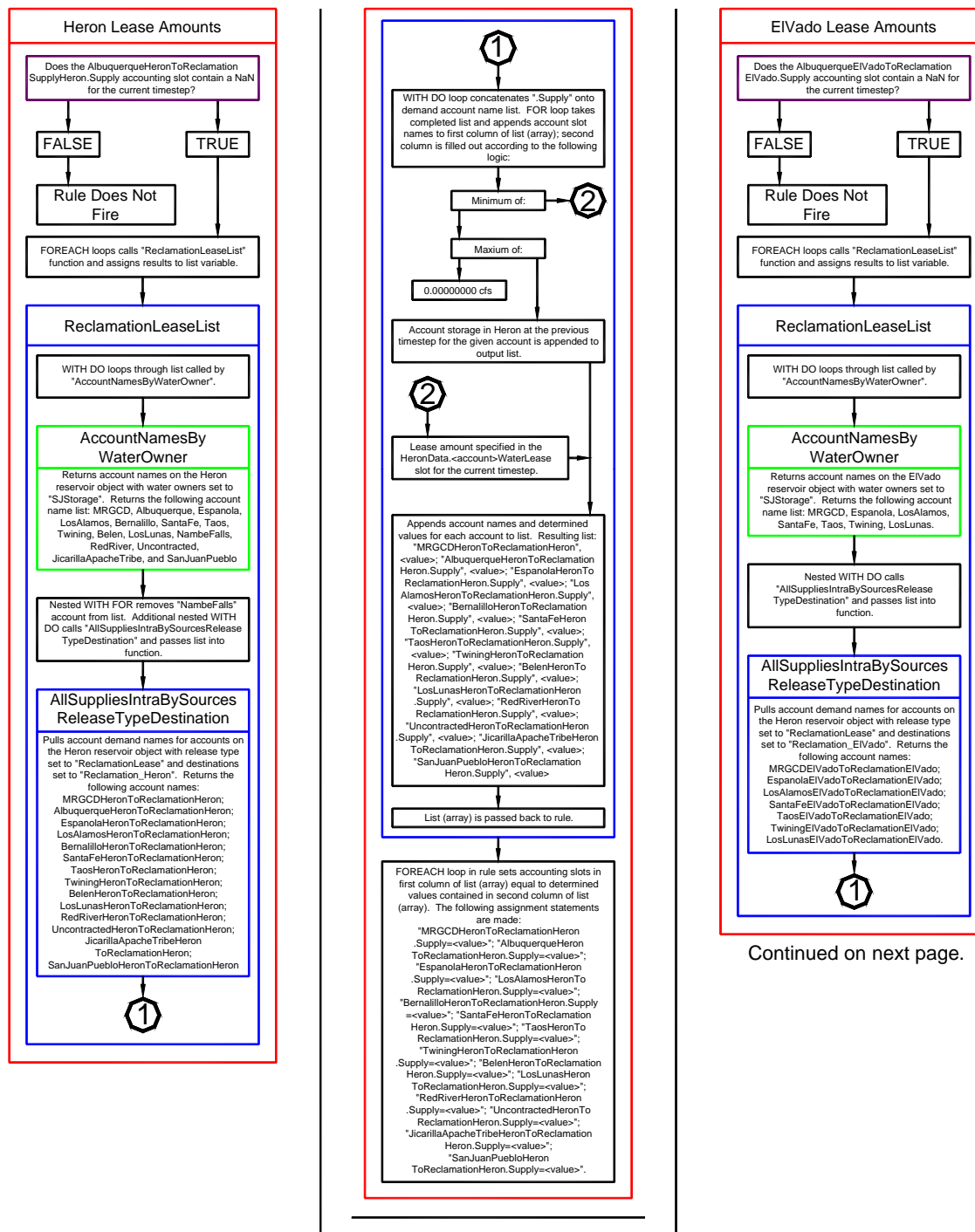


Figure A.9. Flowchart for the Heron Lease Amounts and El Vado Lease Amounts Rules (reference to accounts in the Water Operations Model)

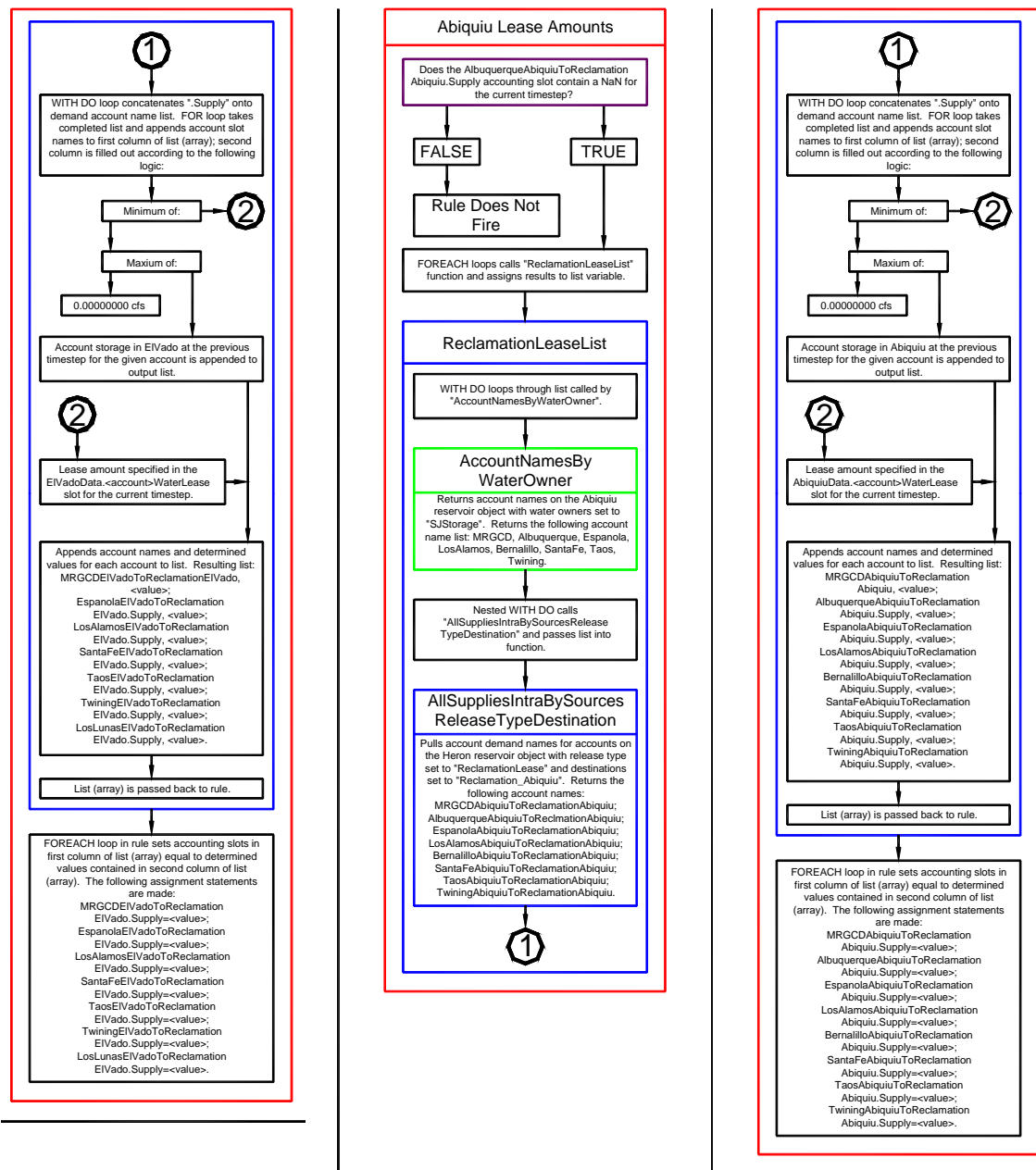


Figure A.10. Flowchart for the El Vado Lease Amounts Rule (continued) and the Abiquiu Lease Amounts Rule (reference to accounts in the Water Operations Model)

#### A.4.4. Set MRGCD Loans

*Explanation:* This rule sets exchanges based on input debt amounts for Reclamation and Albuquerque to payback MRGCD for past loans. Separate Reclamation debts to MRGCD are established to be paid back from Heron, El Vado, and Abiquiu reservoirs (in that order) to payback the total input amount for the current timestep. The destination for the payback (El Vado or Abiquiu Reservoir) is based on the input reservoir priority for the current timestep (reference the HeronData.ReservoirPriority periodic slot). Separate Albuquerque debts to MRGCD are paid back from Heron to El Vado Reservoir or at Abiquiu Reservoir to payback the total input amount for the current timestep. The separate debts are only established based on the available storage for the source account (Reclamation or Albuquerque) at the source reservoir with consideration for any existing debts to be paid back from the source account.

*Rule Execution:* If the value for the MRGCDElVadoAlbuquerqueHeron.Borrow is a NaN for the current timestep, the rule fires. The check against this single exchange set with the rule ensures that the rule only fires once per timestep.

*Rule Logic:* Two FOREACH loops are used in the rule to set the exchange amounts. For the first FOREACH loop, the GetSortedReclamationMRGCDLoans function in the Accounts List utility group is used to create the list of exchange names for separate Reclamation debts to MRGCD and the values to be set, and the GetSortedAlbuquerqueMRGCDLoans function is used to create the list of exchange names for Albuquerque debts to MRGCD and values to be set.

The Reclamation debts are set to be paid back from different source locations (Heron, El Vado, and Abiquiu Reservoirs) to the MRGCD account at either El Vado or Abiquiu reservoir based on input priorities checked with the user-defined ElVadoIsPriority function. Separate exchanges are set to make the payback for the total input amount to the ReclamationFromMRGCDLoan series slot in the HeronData data object for the current timestep. Albuquerque debts are set to be paid back from Heron or Abiquiu Reservoirs to the MRGCD accounts at El Vado or Abiquiu, respectively, to make the payback for the total input amount to the AlbuquerqueFROMMRGCDLoan series slot in the HeronData data object for the current timestep. Refer to Figures A.11 and A.12 for a flowchart depicting the logic in this rule.

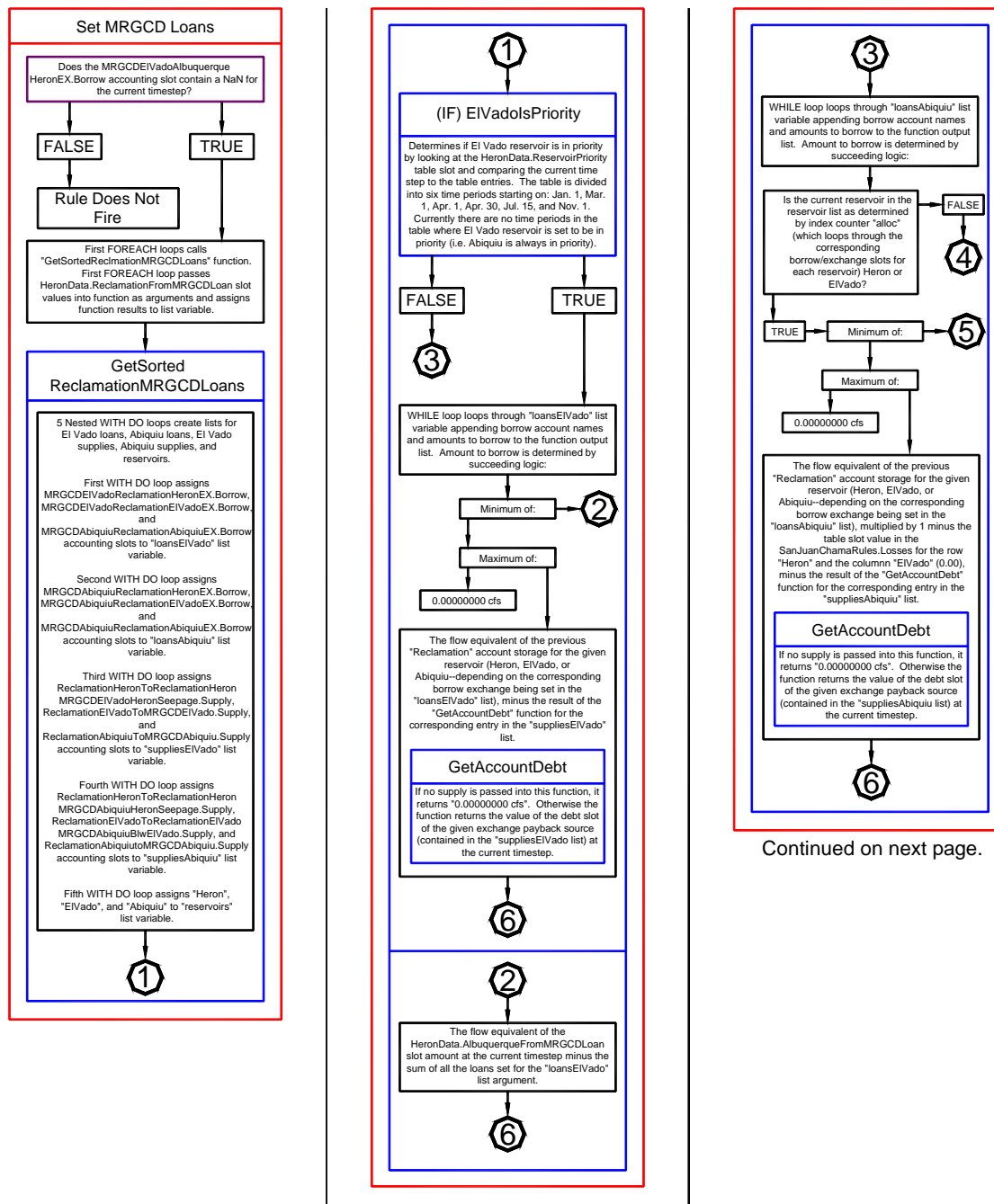


Figure A.11. Flowchart for the Set MRGCD Loans Rule

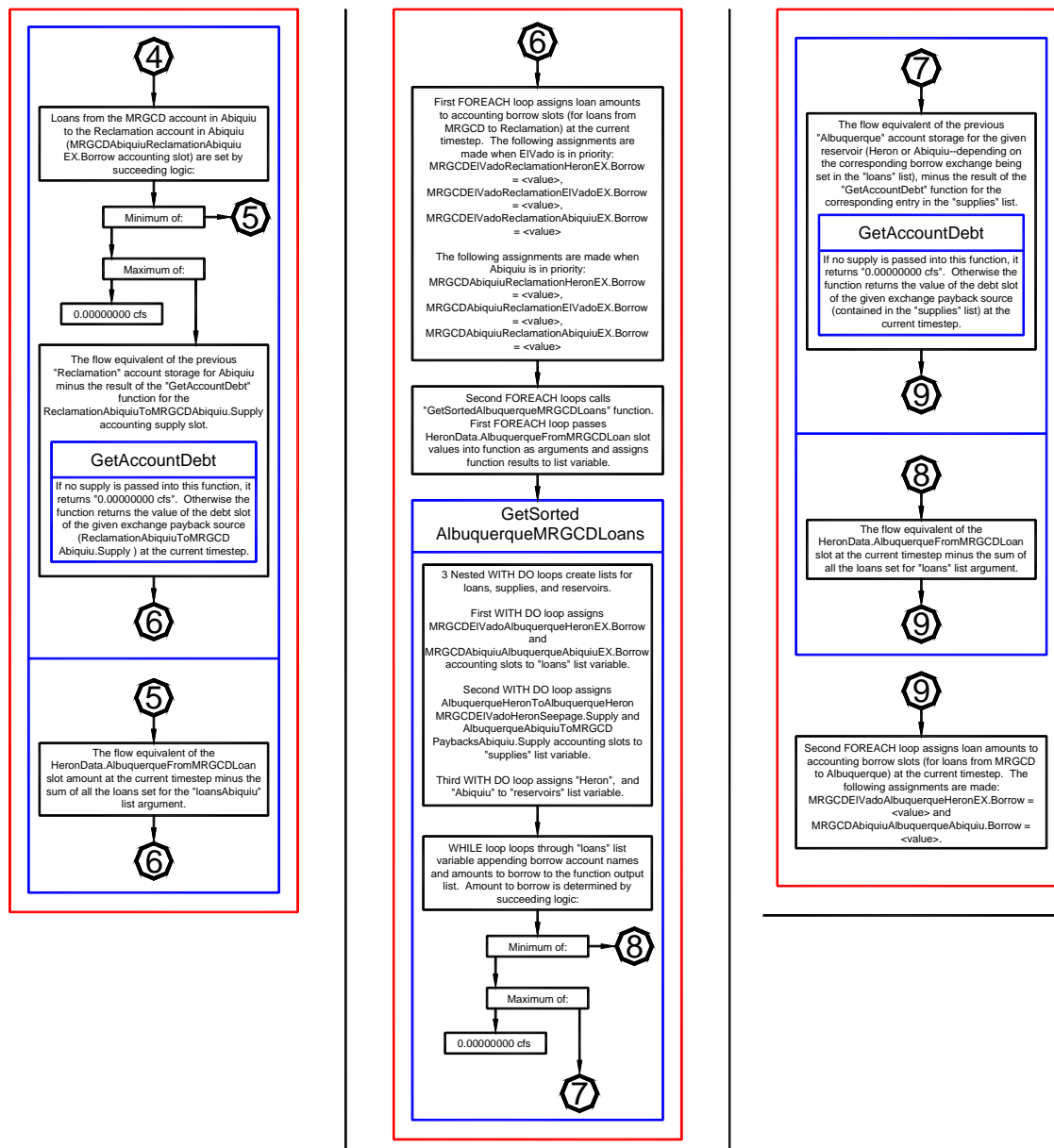


Figure A.12. Flowchart for the Set MRGCD Loans Rule (continued)

#### **A.4.5. Set Reclamation ElVado Abiquiu Payback**

*Explanation:* This rule sets the supplies for Reclamation to transfer water to MRGCD at El Vado and Abiquiu Reservoirs and for Albuquerque to transfer water to MRGCD at Abiquiu Reservoir to payback the debts established with the Set MRGCD Loans rule.

*Rule Execution:* If the value for the `ReclamationElVadoToMRGCDElVado.Supply` is a NaN for the current timestep, the rule fires. This check against one of the supplies set with the rule ensures that the rule only fires once per timestep.

*Rule Logic:* Three assignments are made with this rule. First, if El Vado is the priority destination for the Reclamation payback to MRGCD as checked with the user-defined `ElVadoIsPriority` function, the value for the supply to transfer water from Reclamation's account to MRGCD's account at ElVado Lake is set to the payback debt identified with the user-defined `GetAccountDebt` function in the Exchange Functions utility group which references the predefined `GetPaybackDebt` function. If El Vado Reservoir is not the priority destination, the supply is set to zero. The transfer is limited to the amount that Reclamation has in its storage account. The `ReclamationAbiquiuToMRGCDAbiquiu.Supply` and `AlbuquerqueAbiquiuToMRGCDPaybacksAbiquiu.Supply` are set in the same manner to the established debt for the supply as tracked with exchanges not to exceed the available storage for the source account. Refer to Figure A.13 for a flowchart that depicts the logic used for the three assignments in this rule.

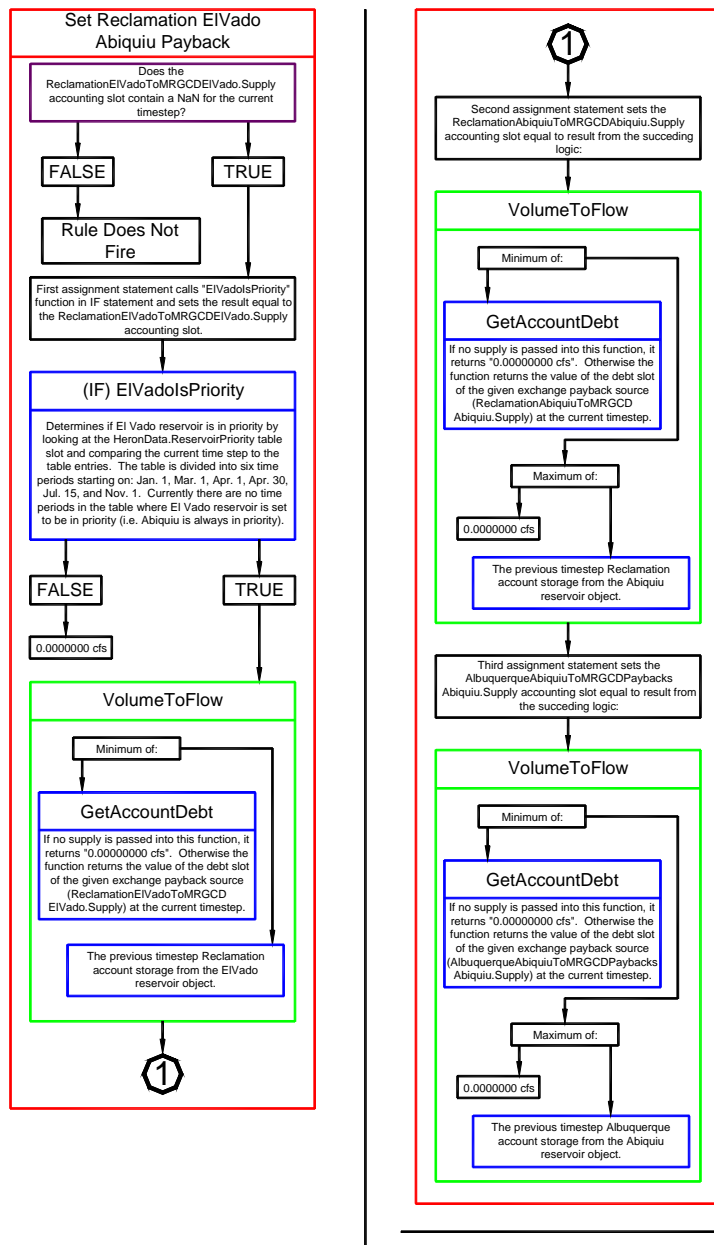


Figure A.13. Flowchart for the Set Reclamation EIVado Abiquiu Payback Rule

#### **A.4.6. Check Delivery Requests Input**

*Explanation:* This rule checks the input delivery requests for contractors for San Juan-Chama Project water to assure that the water is available. The check entails computing the available supply for letter water deliveries as the initial storage for the contractor in account storage at any of the Rio Chama reservoirs plus any expected allocation of San Juan-Chama Project at Heron minus losses to Otowi minus established debt that will need to be paid back. The available water supply is checked against the delivery volume through the end of the first calendar year for a simulation for each contractor, *so the check is only relevant for the first year delivery volume for a multi-year simulation.* Since the number of contractors is reduced for the Planning Model which includes a Combined account, a different set of checks is completed depending on whether the ruleset is being used with the Water Operations or Planning model.

*Rule Execution:* This rule executes at the start timestep.

*Rule Logic:* An IF THEN ELSE statement is used to check against the user defined IsWaterOpsRunNotPlanningRun function to first identify whether the ruleset is being used with the Water Operations Model or Planning Model. A series of IF THEN statements are then used to check the result from the user-defined SumDeliveryRequests function in the Account Summing Functions utility group which is used to compute the volume of the input delivery request for each identified contractor through the end of the first calendar year (or the finish timestep if sooner). The user-defined WaterAvailableForOtowiPayback function in the Account Summing Functions utility group is used to determine the available supply as the sum of the storage the contractor has in all storage accounts on Rio Chama reservoirs plus the expected annual allocation. The available supply is adjusted for the San Juan-Chama loss rates to Otowi, and all established debts for the contractor are subtracted. If the water supply is insufficient to make the delivery through the first year of simulation, the simulation aborts and a message is output to the Diagnostic Output window noting that the input delivery request is greater than the amount of water the contractor has available.

#### **A.5. Forecast Errors**

The rules in this policy group are used to compute a percent forecast error for each month with reference to estimated inflows to El Vado Reservoir. Execution constraints are set to have these rules only fire for the Planning Model. If the rules are turned on, the computed forecast error is then used to incorporate uncertainty in forecasted flows within a simulation. The computed forecast error, not to exceed input maximums, is referenced in other rules including the calculation of a forecasted Otowi flow volume. These rules have been turned off for recent URGWOM applications.



### **A.5.1. RewindRandomFile**

*Explanation:* This rule rewinds the random number file. The seed for the random number generator is constant so that simulations can be regenerated if needed. Refer to the discussion of the predefined ResetRanDev function in the RiverWare online help for further discussion of the random number generator.

*Rule Execution:* This rule executes at the start timestep if a simulation is being completed with the Planning Model (and the rule is turned on).

*Rule Logic:* The function is imbedded in a Print statement, so note that it will not execute if the diagnostics are turned off.

### **A.5.2. CalculatedForecastError**

*Explanation:* This rule computes a forecast error using a random number generator. The forecast error is then used to compute a percent forecast error.

*Rule Execution:* If the ruleset is being used with the Planning Model as determined with a check against NOT IsWaterOpsRunNotPlanningRun function and if the current timestep is the first day of a month and the forecast error for the end of the current month is a NaN as set to the ForecastError time series slot in the ForecastData data object, the rule fires.

*Rule Logic:* The ComputeForecastError function in the ForecastErrorFunctions utility group is used to compute the forecast error using different methods depending on the month (Refer to Figure A.14 for a screen capture of the RiverWare Rule Policy language for this user-defined function). Up to May, the error is computed as a function of input coefficients, the estimated inflow to El Vado Reservoir, the previous forecast error, and a random number. The forecast error in June is set to half the forecast error for May, and the forecast error in July is set to a quarter of the forecast error for June. During the remaining months, it is set to zero. Within the rule, the results from the ComputeForecastError function are then checked against input maximum forecast errors for each month for El Vado Reservoir with consideration for the sign of the error. If the maximum is exceeded, the forecast error is reset to the maximum with consideration for the sign.

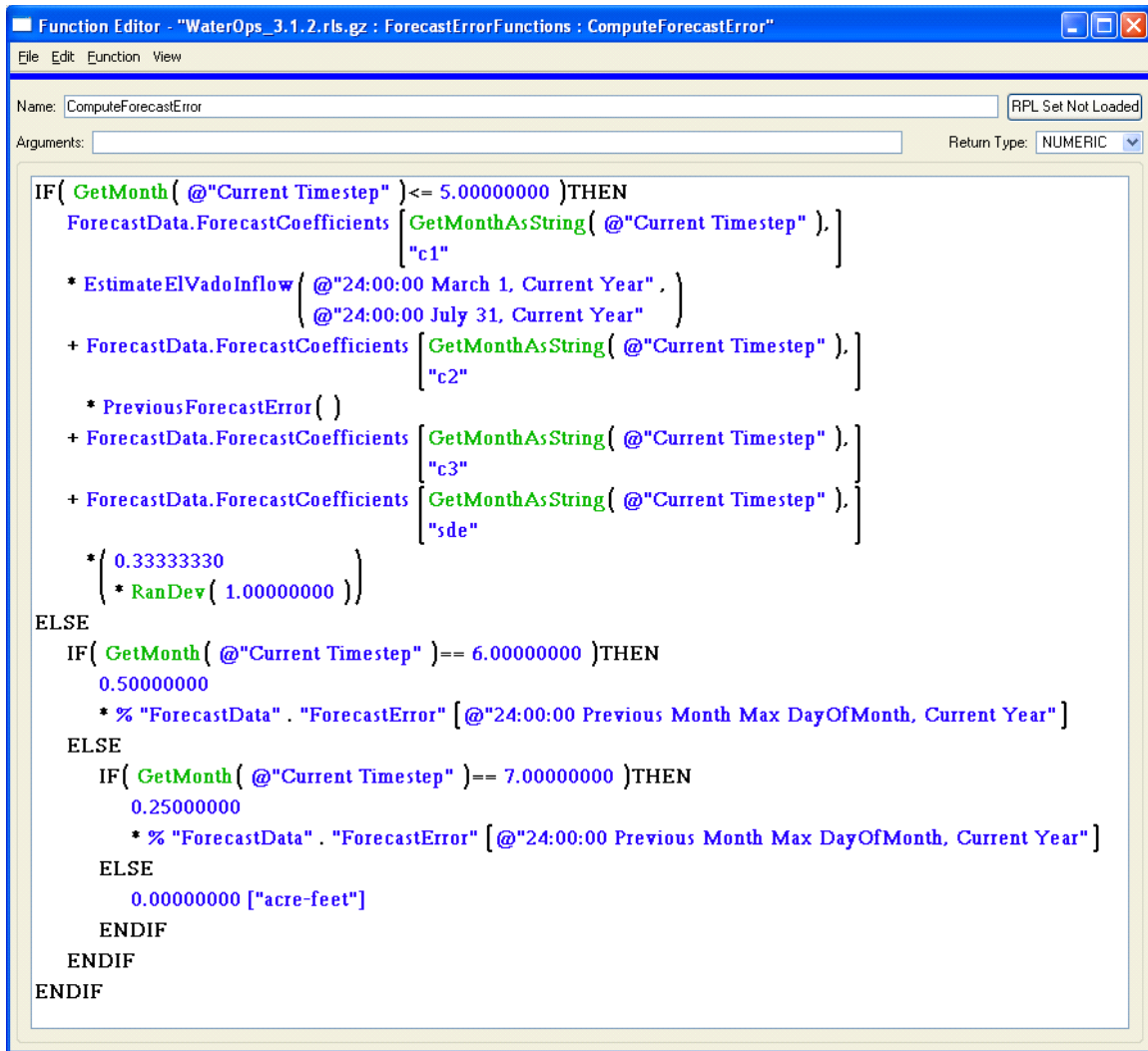


Figure A.14. Rule Policy Language for the ComputeForecastError function used in the CalculatedForecastError Rule

### A.5.3. ForecastErrorPercent

*Explanation:* This rule computes a percent forecast error from the forecast error calculated with the CalculatedForecastError Rule. The percent forecast error is used later when forecasting river flows at Otowi or inflows to Abiquiu Reservoir.

*Rule Execution:* If the ruleset is being used with the Planning Model as determined with a check against NOT IsWaterOpsRunNotPlanningRun function and if the current timestep is the first day of a month and if the percent forecast error for the end of the current month is a NaN as entered in the PercentForecastError time series slot in the ForecastData data object, the rule fires. The second criterion ensures that the rule only fires once per timestep.

*Rule Logic:* The percent forecast error is computed differently depending on the month. Through July, the percent forecast error is the lesser of the forecast error divided by the estimated inflow to El Vado Reservoir and the input maximum percent forecast error with consideration for the sign. If the estimated inflow to El Vado Reservoir is zero, the percent forecast error is set to 0.10. During other months, if it is the first year of the simulation, the percent forecast error is set to 0.04, and if it is not the first year of the simulation, the percent forecast error is set to the lesser of 0.1 or the computed percent forecast error for July of the current year. Refer to Figure A.15 for a screen capture of the RPL for this rule.

```

Rule Editor - "URGWOM_3-5.rls : Forecast Errors : ForecastErrorPercent"
File Edit Rule View
Name: ForecastErrorPercent RPL Set Loaded

ForecastData.PercentForecastError [ EndOfMonth ( ) ]
= IF ( NOT IsNaN ForecastData.ForecastError [ EndOfMonth ( ) ] ) THEN
  IF ( GetMonth ( @"Current Timestep" ) <= 7.0000000 ) THEN
    IF ( EstimateElVadoInflow ( @"24:00:00 March 1, Current Year" ,
      @"24:00:00 July Max DayOfMonth, Current Year" ) != 0.0000000 ["acre-feet"] ) THEN
      IF ( ForecastData.ForecastError [ EndOfMonth ( ) ] > 0.0000000 ["acre-feet"] ) THEN
        Min (
          ForecastData.ForecastError [ EndOfMonth ( ) ] ,
          EstimateElVadoInflow ( @"24:00:00 March 1, Current Year" ,
            @"24:00:00 July Max DayOfMonth, Current Year" )
          ForecastData.MaxPercentForecastError [ 0 ,
            0 ]
        )
      ELSE
        Max (
          ForecastData.ForecastError [ EndOfMonth ( ) ] ,
          EstimateElVadoInflow ( @"24:00:00 March 1, Current Year" ,
            @"24:00:00 July Max DayOfMonth, Current Year" )
          ForecastData.MaxPercentForecastError [ 0 ,
            0 ] * - 1.0000000
        )
      ENDIF
    ELSE
      0.1000000
    ENDIF
  ELSE
    IF ( GetYear ( @"Current Timestep" ) == GetYear ( @"Start Timestep" ) ) THEN
      0.0400000
    ELSE
      Min ( 0.1000000 ,
        ForecastData.PercentForecastError [ @"24:00:00 July Max DayOfMonth, Current Year" ] )
    ENDIF
  ENDIF
ENDIF

Execute Block Only When
NOT IsWaterOpsRunNotPlanningRun ( )
AND IsFirstDayOfMonth ( )
AND IsNaN ForecastData.PercentForecastError [ EndOfMonth ( ) ]

```

Figure A.15. Rule Policy Language for the ForecastErrorPercent Rule

## **A.6. NMCreditsAndDebitsRules**

The rules in this policy group are used to adjust the New Mexico and Colorado Credit for the Compact based on evaporation losses, determine whether the maximum Credit has been reached, input relinquished credits, or whether Elephant Butte is spilling.

### **A.6.1. Set NM Credit Adjustment**

*Explanation:* This rule makes the annual transfer from the Rio Grande account to the NMCredit account at Elephant Butte Reservoir (or from the NMCredit account to the Rio Grande account) to reflect the annual Compact adjustment as computed with the NMCreditAdj expression series slot in the RioGrandeCompact data object. The December 31<sup>st</sup> adjustment is made on January 1<sup>st</sup> of each year (after the first year of simulation).

*Rule Execution:* If it is not the start timestep and the value for the EBUnfilledCredit in the RioGrandeCompact data object for the previous timestep is greater than or equal to zero, the rule fires. (Note that the check against the December 31<sup>st</sup> timestep to make the annual adjustment is completed within the rule as opposed to within the execution constraint. The transfer is set to zero for all other timesteps.)

*Rule Logic:* Two assignments are made with this rule. The GET STRING @INDEX statement is used with the user-defined AllSuppliesIntraByReleaseTypeDestination function in the Account Lists utility group to identify the supply names for both assignments. IF THEN ELSE statements are used to assign values depending on the value for the NMCreditAdj slot in the RioGrandeCompact data object and if the date is December 31. If it is not December 31, the values for both supplies are set to 0.0 cfs. Otherwise, if the value for the NMCreditAdj is greater than or equal to zero, the supply is set to transfer water to the NMCredit account at Elephant Butte Reservoir from the Rio Grande account based on the computed NMCreditAdj value for the previous timestep converted to a flow (i.e. a Compact credit for the year results in a transfer from the Rio Grande account to the Compact credit account). For the second assignment, if the NMCreditAdj is less than zero, the supply is set to transfer water to the Rio Grande storage account from the NMCredit account as the negative of the NMCreditAdj for the previous timestep converted to a flow (i.e. a Compact deficit for the year results in a transfer from the Compact credit account to the Rio Grande account).

### **A.6.2. Zero NM CO Credits**

*Explanation:* If a spill is occurring at Elephant Butte Reservoir, this rule zeroes out the New Mexico and Colorado Compact credits at Elephant Butte Reservoir.

*Rule Execution:* If the value for the ElephantButteSpillSwitch time series slot in the RioGrandeCompact data object for the previous timestep is equal 1.0, the rule fires.

The ElephantButteSpillSwitch is an expression series slot that sets the switch to 1.0 if the EBUfilledCredit expression series slot value is negative. After the switch has been set to 1.0, the value is maintained for the remainder of the calendar year. The EBUfilledCredit slot in the RioGrandeCompact data object is computed as the Elephant Butte Reservoir capacity at the top of conservation minus an input reserve capacity minus the water in storage with any Compact credit (or San Juan-Chama Project water) subtracted. A negative value indicates that storage of native Rio Grande water at Elephant Butte Reservoir has exceeded the limit to indicate a spill.

*Rule Logic:* GET STRING @INDEX statements are used with the user-defined AllSuppliesIntraByReleaseTypeDestination function in the Account Lists utility group to identify the names for the supplies for the four assignment statements. Supplies are set to adjust the amount of water in the NMCredit and the COCredit storage accounts at Elephant Butte Reservoir and correspondingly adjust the amount of water in the Rio Grande storage account at Elephant Butte Reservoir. IF THEN ELSE statements are used set the relevant supplies depending on whether there is a credit or debt.

## **A.7. CO Credit Rules**

The single rule in this policy group is used for an annual transfer from the Colorado credit account to the Rio Grande storage account at Elephant Butte Reservoir, but the Colorado Compact credit/deficit is not actually being input or computed in the latest URGWOM applications.

### **A.7.1. CO Credit**

*Explanation:* This rule sets the supply for an annual transfer of water from the Colorado credit account to the Rio Grande storage account at Elephant Butte Reservoir.

*Rule Execution:* If it is not the start timestep and the supply at Elephant Butte Reservoir to transfer water from the Rio Grande account to the COCredit account is a NaN, the rule fires. (Note that the check against the December 31<sup>st</sup> timestep to make the annual adjustment is completed within the rule as opposed to within the execution constraint. The transfer is set to zero for all other timesteps.)

*Rule Logic:* GET STRING @INDEX statements are used with the user-defined AllSuppliesIntraByReleaseTypeDestination function in the Account Lists utility group to identify the names for the supplies in the two assignment statements. IF THEN ELSE statements are used to set the values for the supplies depending on whether the previous timestep is December 31 and the value in the COCreditDebit time series slot in the RioGrandeCompact data object for the previous timestep. For the first assignment

statement, if that slot value is greater than or equal to zero, the supply to transfer water to the COCredit account at Elephant Butte Reservoir is set to that slot value (i.e. a Compact credit for the year results in a transfer from the Rio Grande account to the Colorado Compact credit account). For the second assignment statement, if the COCreditDebit slot value is less than zero, the supply to transfer water to the Rio Grande account is set to that slot value (i.e. a Compact deficit for the year results in a transfer from the Colorado Compact credit account to the Rio Grande account).

## **A.8. Diversions and Demands**

The rules in this policy group are used to ultimately compute demands for water throughout the Middle Valley and the corresponding required minimum dam releases to meet those demands.

### **A.8.1. SanMarcialElephantButteRiparianLoss**

*Explanation:* This rule computes a loss from the riparian vegetation in the Elephant Butte delta area. The loss in cfs is computed based on the input ET Rate for the current timestep multiplied by a riparian area determined with reference to a lookup table where the riparian area is a function of the Elephant Butte storage (Losses are not computed from the inundated riparian area).

*Rule Execution:* If the rule has not fired successfully for the current timestep as determined with reference to the predefined HasRuleFiredSuccessfully function, the rule fires.

*Rule Logic:* A FOREACH loop is used to set the loss for the current timestep through the current timestep plus seven days. (The assignments are set into the future in case the values are needed for hypothetical simulation from Cochiti to the referenced objects.)

The value in the ReachRiparianET series slot in the SanMarcialToElephantButteRiparianLossCalc data object is set to the minimum of the previous Inflow to the SanMarcialConfluenceToElephantButte reach object above Elephant Butte Reservoir and the computed loss. The computed loss is calculated based on the value in the ETRate series slot in the SanMarcialToElephantButteRiparianLossCalc data object for the current timestep and a riparian area determined using the predefined TableInterpolation function with reference to the Elephant Butte pool elevation at the previous timestep and the table input to the RiparianAreaReservoirElevationTable table slot in that same data object.

### **A.8.2. SanAcaciaSocorroMainCanalDiversionRequest**

*Explanation:* This rule sets the diversion requested to the Socorro Main Canal at the San Acacia diversion based on an input diversion schedule minus the through flow from the

Unit 7 drain in the MRGCD system. Flows from the Unit 7 drain contribute toward meeting the need at the Socorro main canal, and diversions from the river are curtailed accordingly. The diversion requested values for San Acacia are set separately from the Cochiti, Angostura, and Isleta diversions due to the unique dependency at San Acacia on the Unit 7 through flow.

*Rule Execution:* If the rule has not fired successfully for the current timestep as determined with reference to the predefined HasRuleFiredSuccessfully function, the rule fires.

*Rule Logic:* A FOREACH loop is used to set the value for the Diversion Requested series slot for the Canal Div water user on the SanAcaciaDiversions aggregate diversion site object from the current timestep through the current timestep plus one day (i.e. the approximate travel time from Abiquiu Dam to Cochiti Lake) plus the travel time from Cochiti Dam to San Marcial as input to the ApproxNoOfDaysDS table slot in the CochitiData data object. (The assignment is set into the future as needed for the hypothetical simulation to estimate flows needed for targets.) The diversion is set to the value input to the CanalDiv series slot in the MiddleRioGrandeDiversionData data object minus the value in the Total Outflow series slot for the previous timestep in the DrainUnit7Return aggregate distribution canal object. The associated value for the Depletion Requested series slot for the Canal Div water user on the SanAcaciaDiversions aggregate diversion site object is set to zero.

### **A.8.3. SanAcaciaLFCCDiversionsRequest**

*Explanation:* This rule sets the diversion requested values for the Low Flow Conveyance Channel (LFCC) at the San Acacia diversion for the entire simulation period based on an input diversion schedule. Diversions to the LFCC have been set to zero for all recent URGWOM applications.

*Rule Execution:* If it is the Start Timestep, the rule fires.

*Rule Logic:* A FOREACH loop is used to set the the value for the Diversion Requested series slot for the LFCCDiversions water user on the SanAcaciaDiversions aggregate diversion site object from the current timestep through the finish timestep. Diversions are set to the values input to the LFCCDiversions series slot in the MiddleRioGrandeDiversionData data object for the corresponding date. The associated values for the Depletion Requested series slot for the LFCCDiversions water user on the SanAcaciaDiversions aggregate diversion site object are all set to zero.

#### **A.8.4. CentralWastewayCalcs**

*Explanation:* This rule sets the MRGCD return to the river from the Central wasteway and the flow through the Atrisco siphon. Policy is coded for setting the flow through the wasteway and the siphon during the irrigation season versus the non-irrigation season.

*Rule Execution:* If the rule has not fired successfully for the current timestep as determined with reference to the predefined HasRuleFiredSuccessfully function, the rule fires.

*Rule Logic:* A FOREACH loop is used to set the value for the Diversion Request series slot for the CentralWasteway and AtriscoSyphon diversion objects from the current timestep through the current timestep plus one day (i.e. the approximate travel time from Abiquiu Dam to Cochiti Lake) plus the travel time from Cochiti Dam to San Marcial as input to the ApproxNoOfDaysDS table slot in the CochitiData data object. (The assignment is set into the future as needed for the hypothetical simulation to estimate flows needed for targets.) During the irrigation season (March 1 – October 31), the flow through the Central wasteway is set to the flow at that location in the MRGCD system, represented by the Outflow value from the CentralEastSideLag reach object, minus 180 cfs. During the non-irrigation season, all flow in the system is returned through the wasteway. During the irrigation season, the flow diverted through the Atrisco siphon is set to 120 cfs if the Outflow value from the CentralEastSideLag reach object is greater than or equal to 180 cfs. Alternatively, the flow through the siphon is set to the flow in the system \* 0.67 during the irrigation season. During the non-irrigation season, no flow is diverted through the Atrisco siphon.

#### **A.8.5. LFCCPumpingRequested**

*Explanation:* This rule sets the pumping rates from the Low Flow Conveyance Channel (LFCC) at the Neil Cupp, North Boundary of the Bosque del Apache National Wildlife Refuge, and South Boundary sites. Policy is coded for pumping water from the LFCC to the river to manage recession after the runoff or prevent river drying. Water that seeps into the Low Flow Conveyance Channel is pumped to the river where pumping begins based on input river flow triggers. After pumping has initiated at a site, pumping will continue for a minimum of one week and until a threshold flow at San Acacia has been exceeded. Pumping will cease for the year at each site after the specific dates input for each site.

*Rule Execution:* If the rule has not fired successfully for the current timestep as determined with reference to the predefined HasRuleFiredSuccessfully function, the rule fires.



*Rule Logic:* A FOREACH loop is used to set the value for the Diversion Request series slots for the Neil Cupp, North Boundary, and South Boundary diversion objects from the current timestep through the current timestep plus one day (i.e. the approximate travel time from Abiquiu Dam to Cochiti Lake) plus the travel time from Cochiti Dam to San Marcial as input to the ApproxNoOfDaysDS table slot in the CochitiData data object. (The assignment is set into the future as needed for the hypothetical simulation to estimate flows needed for targets.)

The user-defined LFCCPumping function in the Silvery Minnow Functions utility group is referenced for setting the Diversion Request values to represent the pumping at each site. A series of IF THEN ELSE statements are included to check the conditions for setting the diversion. If the current timestep is after the Start Timestep and before the input date to shutdown pumping for that site for the year as input to the DateToShutDownForYear table slot in the LFCCPumpingTriggers data object and as checked with the user-defined BeforeDateToShutDownLFCCPumpsForYear function, a pumping rate is set.

If the hydrology year type as checked with the user-defined HydrologyYearType function (reference the discussion for the Hydrology Year Type rule) is Dry and the flow at San Acacia is less than the input trigger flow to start pumping at that site as input to the Dry table slot in the LFCCPumpingTriggers data object, the function result is set to the value for the PumpingRate for that site in that same table slot. A similar check is then completed if the hydrology year type is Normal. If the hydrology year type is Wet and the current timestep is within the “winter target season” for wet targets as checked with the user-defined IsWetSanMarcialWinterTargetSeason function and the flow at San Acacia is less than the input trigger to start pumping at that site for wet years, the function result is set to value for the WinterPumpingRate in the Wet table slot in the LFCCPumping triggers data object. Alternatively, if the hydrology year type is Wet and the current timestep is NOT within the winter target season for wet targets and the flow at San Acacia is less than the input trigger to start pumping at that site for wet years, the function result is set to value for the SummerPumpingRate in the Wet table slot in the LFCCPumping triggers data object.

If pumping occurred at the previous timestep for the site as checked with the user-defined LFCCPumpingOnPreviousDay function and pumping has not continued for at least a week as checked with NOT LFCCPumpingForAtLeastAWeek function or the flow at San Acacia has not exceeded the trigger flow within the past week to shutdown pumping as checked with the SATriggerForLFCCPumpingShutdownExceededAWeek function, the function result is set to the previous pumping rate (i.e. the previous set pumping rate is maintained). Otherwise, the pumping rate is set to zero.

#### **A.8.6. CompactVIIUsableStorage**

*Explanation:* This rule computes the “usable storage” to be referenced by the CompactVIIUsableStorage rule when identifying whether the stipulations of Article VII

of the Compact are in effect. Usable storage is computed as the total storage at Elephant Butte and Caballo Reservoirs minus any credit water for New Mexico and Colorado and minus San Juan-Chama Project water in the Albuquerque account. Note that the storage in the three accounts is only subtracted if the account storage is positive (i.e. any tracked Compact *debt* as negative account storage is not considered in the calculation). Also, the usable storage is immediately adjusted for any relinquished credit (i.e. the transfer of water from the NMCredit account to Rio Grande storage as a result of relinquished credits).

*Rule Execution:* If the value in the CompactVIIUsableWater time series slot in the RioGrandeCompact data object is a NaN for the current timestep, the rule fires.

*Rule Logic:* The value for the CompactVIIUsableWater time series slot in the RioGrandeCompact data object for the current timestep is set using an assignment statement to the value for the Storage at Elephant Butte Reservoir at the previous timestep as referenced with the user-defined PreviousStorage function in the Get Data Functions utility group minus the storage in the Albuquerque, COCredit, and NMCredit storage accounts at Elephant Butte Reservoir on December 31<sup>st</sup> of the previous year or the initial timestep, if the corresponding account storage on that date is greater than 0.0 acre-ft as assured with the predefined Max function plus any relinquished credit in the RelinquishedNMCredits series slot in the El Vado Data data object if the current timestep is after the relinquishment date as identified with the user-defined RelinquishmentDate function plus the value for the Storage at Caballo Reservoir at the previous timestep as referenced with the user-defined PreviousStorage function.

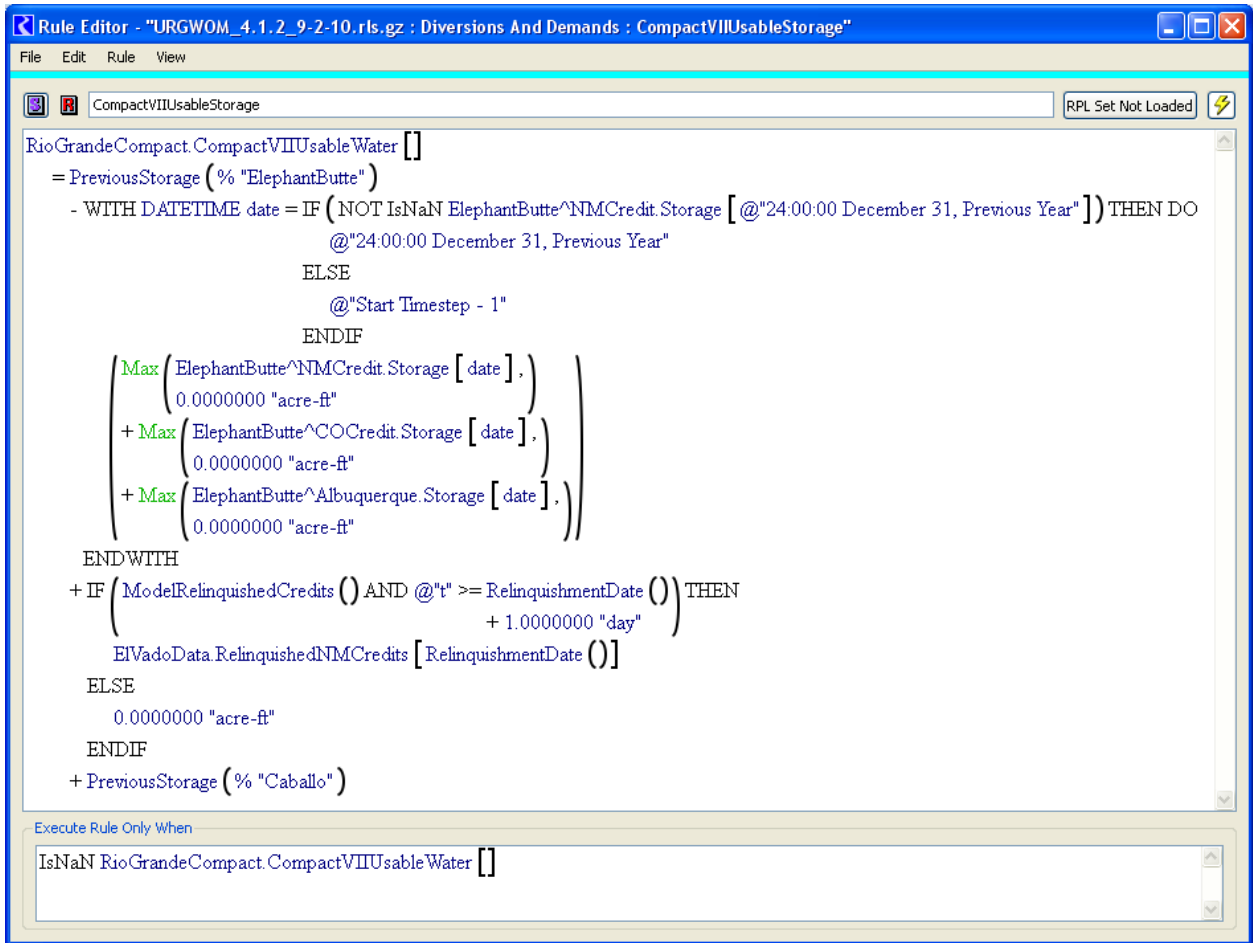


Figure A.16. Rule Policy Language for the CompactVIIUsableStorage Rule

### A.8.7. SetCompactVIISwitch

*Explanation:* This rule sets a switch that identifies whether the rules stipulated in Article VII of the Compact apply which depend on whether the usable storage as determined with the CompactVIIUsableStorage Rule is less than a minimum storage of 400,000 acre-ft.

*Rule Execution:* If the value in the ArticleVIISwitch time series slot in the RioGrandeCompact data object is a NaN for the current timestep, the rule fires.

*Rule Logic:* If the value for the CompactVIIUsableWater time series slot in the RioGrandeCompact data object for the previous timestep is less than the minimum storage input to the MinimumStorage table slot in the RioGrandeCompact data object (400,000 acre-ft), the value for the switch is set to 1.0. Otherwise, the value is set to zero. The result is recorded to the ArticleVIISwitch time series slot in the RioGrandeCompact data object for the current timestep. The input minimum storage is referenced with the user-defined CompactMinStorage function in the Get Data Functions utility group. An

identical assignment statement is included to be applied at the Start Timestep for specifically setting the value for the Initial Timestep if a value was not input.

### **A.8.8. ComputeOtowForecast**

*Explanation:* This rule is used to compute a forecasted flow at Otowi for the period from March through July. The Otowi forecast is used later to identify the type of year when setting flow targets.

*Rule Execution:* If it is the start timestep and before May 1 or the first day of a month on or before May 1 of the current year, the rule fires.

*Rule Logic:* If the OtowiForecast time series slot in the Indian data object for the current year is not input, it is set using the user-defined RouteRioGrandeToOtow function in the Forecast Functions utility group to compute the expected flow at Otowi. An IF THEN ELSE statement is used to apply a PercentForecastError (Refer to the ForecastErrorPercent Rule) if a value was set in the PercentForecastError time series slot in the ForecastData data object for the current month.

The RouteRioGrandeToOtow function references several other functions in the Forecast Functions utility group. All of these functions use the predefined SumFlowsToVolume function to compute the volume of flow from a particular source that would be expected at Otowi for the period from March 1 through July 31. Each of these functions considers percent losses for the forecast period for the appropriate separate designated reaches from the location of the inflow to Otowi. The user-defined AverageForecastPeriodLoss function in the Forecast functions utility group is used to reference the percent losses, as a function of a reach identifier, which are input to the AverageForecastPeriodLosses table slot in the ForecastData data object.

### **A.8.9. Hydrology Year Type**

*Explanation:* This rule is used to set a trigger for the current timestep to 1, 2, or 3 to identify whether the year is classified as Dry, Normal, or Wet, respectively. The result is used later when downstream target flows are determined. These year classifications are established as needed for defining flow targets per the Biological Opinion. Note that the year classification is checked at the first day of each month through May 1<sup>st</sup>, and the year classification as of May 1<sup>st</sup> is maintained for the remainder of the calendar year.

*Rule Execution:* If the value for the HydrologyYearType time series slot in the MiddleValleyDemands data object is a NaN for the current timestep, the rule fires.

*Rule Logic:* The HydrologyYearType time series slot in the MiddleValleyDemands data object is set using IF THEN ELSE statements. If the current timestep is the Start Timestep or the first day of a month on or before May 1, the following criteria are

checked for potentially adjusting the hydrology year type. Otherwise, the previous established year classification is maintained.

If the stipulations in Article VII of the Compact apply, the value is set to 1.0 to identify that it is a Dry year. Otherwise, the OtowiForecast for the current year computed with the ComputeOtowiForecast Rule is checked against the input AverageOtowiForecast in the Indian data object multiplied by the corresponding forecast factor entered in the ForecastFactorsForHydYearType table slot in the MiddleValleyDemands data object. If the forecast is less than or equal to the average forecast times the Dry forecast factor, the value for the HydrologyYearType slot is set to 1.0 to identify that it is a Dry year. If the Otowi forecast is greater than the average forecast times the Wet forecast factor, the value is set to 3.0 to identify that it is a Wet year. If none of the preceding criteria applies, the value is set to 2.0 to identify that is a Normal year.

#### **A.8.10. SetMinTargetsAtStart**

*Explanation:* This rule is used to identify the downstream target flows at Central, Isleta, San Acacia, and San Marcial at the Start Timestep based on the hydrology year type and with consideration for the input adjustment factor.

*Rule Execution:* This rule executes if the current timestep is the Start Timestep.

*Rule Logic:* A FOREACH loop is used to record the downstream targets flows for four time series slots in the MiddleValleyDemands data object: MinTargetForCentral, MinTargetForIsleta, MinTargetForSanMarcial, and MinTargetForSanAcacia. An internal FOREACH loop is used to make assignments for each slot for the current timestep through the current timestep plus one day (i.e. the approximate travel time from Abiquiu Dam to Cochiti Lake) plus the travel time from Cochiti Dam to the corresponding target location as input to the ApproxNoOfDaysDS table slot in the CochitiData data object.

The user-defined MinTargetFlow function in the Get Data Functions utility group is used to compute the target flow as a function of the date, hydrology year type, and target flow adjustment factor. The target flows for each location based on the date and hydrology year type are input to the MinTargetFlows periodic slot in the MiddleValleyDemands data object, and the target flow adjustment factor for each target location and each month is input to the MinTargetFlowsSafetyFactor periodic slot in the MiddleValleyDemands data object. An identical calculation is completed to also assign the values to series slots that end in “TargetNoOtowiCutoff” in the MiddleValleyDemands data object for each target location (The side calculations are completed to track what targets would be if targets are set to zero as part of simulated proposed action to not use targets after a year-to-date threshold flow volume at Otowi is reached).

### **A.8.11. MinIsletaSanAcaciaSanMarcialFlowTargets**

*Explanation:* This rule is used to set the downstream flow targets at Isleta, San Acacia, and San Marcial based on the hydrology year type and with consideration for the input adjustment factor. (The target for Central is set separately to prevent conflicts with the potential alternate policy for resetting Central targets for recruitment or overbank flows as a result of Cochiti deviations.) Targets are actually set into the future based on the travel time from Abiquiu Dam to the target location. Targets are set to zero after the year-to-date Otowi flow volume has exceeded the input threshold volume for conserving supplemental leased San Juan-Chama Project water and if there is no Emergency Drought water available (i.e. there is no storage in the SupplementalESA account at El Vado Reservoir). Policy for this proposed action of conserving supplemental leased San Juan-Chama Project in wetter years may be turned off by inputting a very high threshold Otowi flow volume.

*Rule Execution:* This rule executes every timestep. The value in the MinTargetForSanMarcial series slot in the Middle Valley Demands data object is checked to see if the value is a NaN (i.e. a value has not already been set) to assure the rule only fires once each timestep.

*Rule Logic:* A FOREACH loop is used to record the downstream flow targets for the timestep equal to the current timestep plus one timestep (the approximate travel time from Abiquiu Dam to Cochiti) plus the travel time from Cochiti to the target location for the following three time series slots in the MiddleValleyDemands data object: MinTargerForIsleta, MinTargetForSanAcacia, and MinTargetForSanMarcial. If the year-to-date modeled flow volume in the Gage Outflow series slot for the Otowi stream gage object is less than the input value in the ThresholdOtowiVolumeHoldLeaseWater scalar slot in the MiddleValleyDemands data object or the storage in the SupplementalESA account at El Vado Reservoir at the previous timestep is greater than zero, the target is determined with reference to the user-defined MinTargetFlow function in the Get Data Functions utility group as a function of the date, hydrology year type, and target flow adjustment factor. The target flows for each location, based on date and hydrology year type, are input to the MinTargetFlows periodic slot in the MiddleValleyDemands data object, and the target flow adjustment factor for each target location and each month is input to the MinTargetFlowsSafetyFactor periodic slot in the MiddleValleyDemands data object. If the condition is not satisfied, the target is set to zero.

An outer IF THEN ELSE statement is included such that the year-to-date Otowi flow volume is not checked if the current timestep is the Start Timestep or January 1. An additional assignment statement is also included to record what the targets would be if the values were not reset to zero after the year-to-date Otowi flow volume exceeds the threshold volume. These assignments are made to three series slots for the target locations that end with “TargetNoOtowiCutoff” in the Middle Valley Demands data object.

## A.8.12. MinCentralFlow Target

*Explanation:* This rule is used to set the downstream flow target at Central based on the hydrology year type and with consideration for the input adjustment factor. (The target for Central is set separately to prevent conflicts with the potential alternate policy for resetting Central targets for recruitment or overbank flows as a result of Cochiti deviations.) The target is actually set into the future based on the travel time from Abiquiu Dam to Central. The target may be set to zero after the year-to-date Otowi flow volume has exceeded the input threshold volume for conserving supplemental leased San Juan-Chama Project water and if there is no Emergency Drought water available (i.e. there is no storage in the SupplementalESA account at El Vado Reservoir). Policy for this proposed action of conserving supplemental leased San Juan-Chama Project in wetter years may be turned off by inputting a very high threshold Otowi flow volume. Note that the rule for resetting the Central targets for Cochiti deviations is higher priority and targets will be subsequently be changed for deviations if deviations are implemented.

*Rule Execution:* This rule executes every timestep. The value in the MinTargetForCentral series slot in the Middle Valley Demands data object is checked to see if the value is a NaN (i.e. a value has not already been set) to assure the rule only fires once each timestep.

*Rule Logic:* A FOREACH loop is used to record the downstream flow target for the timestep equal to the current timestep plus one timestep (the approximate travel time from Abiquiu Dam to Cochiti) plus the travel time from Cochiti to Central for the MinTargetForCentral time series slot in the MiddleValleyDemands data object. If the year-to-date modeled flow volume in the Gage Outflow series slot for the Otowi stream gage object is less than the input value in the ThresholdOtowiVolumeHoldLeaseWater scalar slot in the MiddleValleyDemands data object or the storage in the SupplementalESA account at El Vado Reservoir at the previous timestep is greater than zero, the target is determined with reference to the user-defined MinTargetFlow function in the Get Data Functions utility group as a function of the date, hydrology year type, and target flow adjustment factor. The target flow, based on the date and hydrology year type, is input to the MinTargetFlows periodic slot in the MiddleValleyDemands data object, and the target flow adjustment factor for each target location and each month is input to the MinTargetFlowsSafetyFactor periodic slot in the MiddleValleyDemands data object. If the condition is not satisfied, the target is set to zero.

An outer IF THEN ELSE statement is included such that the year-to-date Otowi flow volume is not checked if the current timestep is the Start Timestep or January 1. An additional assignment statement is also included to record what the targets would be if the values were not reset to zero after the year-to-date Otowi flow volume exceeds the threshold volume. This assignment is made to the SideCalcCentralTargetNoOtowiCutoff series slot in the Middle Valley Demands data object.

### A.8.13. ResetIsletaSanAcaciaSanMarcialTargetsForStepDown

*Explanation:* Targets may be reset in URGWOM for discretionary operations as conducted under the Biological Opinion (Service, 2003) which entail using supplemental water to manage the recession after the runoff and control the rate of drying after river rewetting for minnow salvage. Policy for representing discretionary operations entails implementing a longer step down in targets at the end of the runoff and shorter step downs in targets thereafter following each river rewetting event. Note that a step down in targets as needed after the continuous flow requirement if the runoff ends before the continuous flow requirement is over would be represented separately and would need to be included in the input target table.

This rule resets the targets for a 30-day step down to manage recession at the first occurrence of river drying. The definition of drying and the shape of the step down in targets are set based on input threshold flows and target values. Any input adjustment factor is applied to the computed step down in targets. Targets are not set for discretionary operations if the threshold year-to-date Otowi flow volume for conserving leased San Juan-Chama Project water has been exceeded and there is no Emergency Drought water. The logic used in this rule for potentially resetting the targets originally set with the MinIsletaSanAcaciaSanMarcialFlowTargets rule is depicted by the flowchart in Figure A.17.

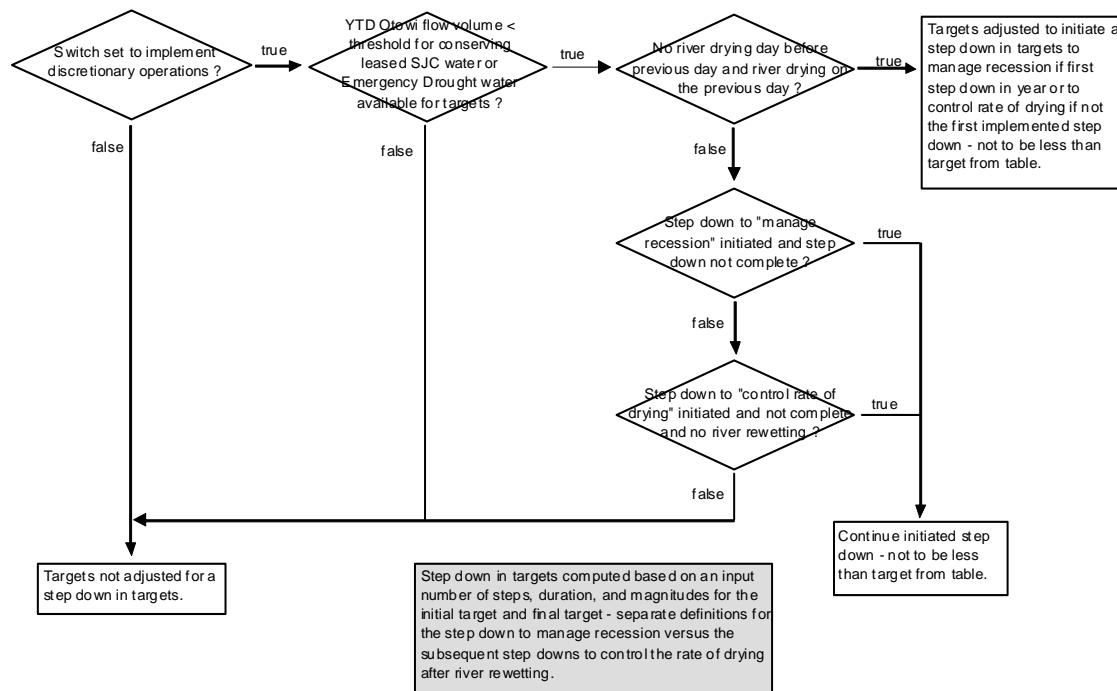


Figure A.17. Flow Chart Depicting Logic for Establishing Step Downs in Targets



*Rule Execution:* This rule fires if the rule has not fired successfully for the current timestep as determined with reference to the predefined HasRuleFiredSuccessfully function and a switch is set to implement the operation based on an input value greater than zero in the TriggerImplementStepDownInTargets scalar slot in the MiddleValleyDemands data object and river drying has initiated as checked with the user-defined TimeToStartStepDownInTargets function or an initiated step down in targets needs to continue as checked with the TimeToContinueStepDownInTargets function in the Silvery Minnow Functions utility group.

*Rule Logic:* A FOREACH loop is used to reset the downstream flow targets for the timestep equal to the current timestep plus one timestep (the approximate travel time from Abiquiu Dam to Cochiti) plus the travel time from Cochiti to the target location for the following three time series slots in the MiddleValleyDemands data object: MinTargetForIsleta, MinTargetForSanAcacia, and MinTargetForSanMarcial. If the year-to-date modeled flow volume in the Gage Outflow series slot for the Otowi stream gage object is less than the input value in the ThresholdOtowiVolumeHoldLeaseWater scalar slot in the MiddleValleyDemands data object or the storage in the SupplementalESA account at El Vado Reservoir at the previous timestep is greater than zero, the target is computed using the user-defined ComptueTargetInStepDown function based on the input values for the final target in the step down, initial target in the step down, number of steps, and duration of the step down. The adjustment factor is then applied to the computed target. If the condition is not satisfied, the target is set to the original table value with reference to the user-defined MinTargetFlow function.

An outer IF THEN ELSE statement is included such that the year-to-date Otowi flow volume is not checked if the current timestep is the Start Timestep or January 1. An additional assignment statement is also included using similar logic except without the check against the year-to-date Otowi flow volume to record what the targets would be if the values were not reset to zero after the year-to-date Otowi flow volume exceeds the threshold volume. These assignments are made to three time series slots for the target locations that end with “TargetNoOtowiCutoff” in the MiddleValleyDemands data object. Also, another assignment statement is included to set the value in the StepDownImplementedCounter series slot in the MiddleValleyDemands data object to track the number of timesteps since the step down was implemented as needed for computing the target during the step down period.

#### **A.8.14. WriteNewMinTargetsForOverbankHydrographKnowledge**

*Explanation:* This rule is used to reset the downstream flow targets at Central with flows to provide an “overbank” hydrograph if Cochiti deviations are implemented to provide overbank flows (A separate rule is used if criteria are satisfied to provide recruitment flows as opposed to overbank flows). The targets for the entire overbank hydrograph are set on the date to begin storage at Cochiti Lake for deviations, as input, or a computed date if no date was input. The computed date is based on an input number of days prior

to the estimated time for the peak inflow to Cochiti. Deviations are not implemented if the current timestep is after the last year that deviations are implemented as input.

*Rule Execution:* The rule files if a switch is set to use an input date for Cochiti deviations as a value set to the UsePresetDayForCochitiDeviationsStorage that is greater than zero checked with the user-defined UsePresetDayToStartCochitiDeviationsStorage function and the current timestep is equal to the timestep to begin storage for deviations as input to the PresetDayForCochitiDeviationsStorage table slot in the MiddleValleyDemands data object or if the current timestep is equal to the computed date to begin storage for deviations computed as the time of the peak inflow to Cochiti Lake offset by the number of days input to the DaysOfCochitiDeviationsStorageBeforePeak scalar slot in the MiddleValleyDemands data object and the current timestep is less than or equal to the value input to the LastYearCochitiDeviationsAuthorized table slot in the MiddleValleyDemands data object and the rule has not fired successfully for the current timestep as determined with reference to the predefined HasRuleFiredSuccessfully function.

*Rule Logic:* A FOREACH loop is used to reset targets at Central as set to the MinTargetForCentral series slot in the MiddleValleyDemands data object for the list of dates and associated targets created with the user-defined EnvironmentalMinRelease function in the RecruitmentAndOverbankFunctions utility group. This EnvironmentalMinRelease function identifies the list of targets to provide overbank flows from the EnvironmentalTargets table slot in the MiddleValleyDemands data object and the associated dates for the assignment are set to begin four days before the estimated date for the peak inflow to Cochiti as determined with the user-defined CochitiMaxInflowDate function and continue until 25 days after the date of the peak. (Note that the targets may be reset back to the original table targets before the end of the 30-day hydrograph if storage for deviations at Cochiti Lake drops below a threshold volume before the end of the deviations period. Refer to the EndTargetsForOverbankOrRecruitment rule.)

Within the rule, targets are reset to provide overbank flows if the Otowi forecast is within the range for conducting deviations to provide *overbank* flows as checked with the user-defined IsOverbankForecastRange function and the projected peak inflow to Cochiti is within the range for conducting deviations to provide *overbank* flows as input to the EnvironmentalMinMaxPeakFlows table slot in the MiddleValleyDemands data object or the Otowi forecast is within the range for conducting deviations to provide *recruitment* flows but the projected peak inflow to Cochiti is within the range for conducting deviations to provide *overbank* flows as input to the EnvironmentalMinMaxPeakFlows table slot.

An additional assignment statement is also included using the exact same logic to reset the targets for the SideCalcCentralTargetNoOtowiCutoff series slot in the MiddleValleyDemands data object. Cochiti deviations are not impacted by the policy for conserving supplemental leased San Juan-Chama Project water if a threshold year-to-date

Otowi flow volume is exceeded, so the targets are recorded to this slot regardless of the year-to-date Otowi flow volume.

Refer to Figure A.18 for a flowchart that depicts the policy for Cochiti deviations as represented by the WriteNewMinTargetsForOverbankHydrographKnowledge, WriteNewMinTargetsForRecruitmentHydrographKnowledge, and ComputeCochitiRGConservationSpaceAvailableHydrographKnowledge rules with reference to input values in the current model.

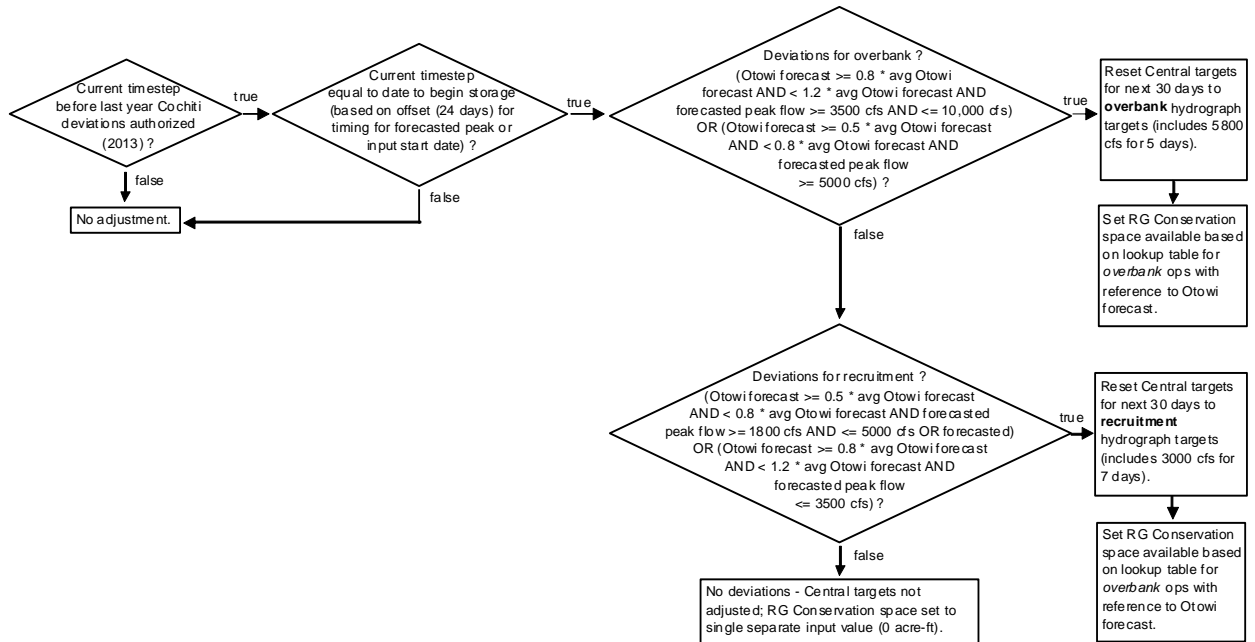


Figure A.18. Flowchart for Implementing Cochiti Deviations

### A.8.15. WriteNewMinTargetsForRecruitmentHydrographKnowledge

**Explanation:** This rule is used to reset the downstream flow targets at Central with flows to provide a “recruitment” hydrograph if Cochiti deviations are implemented to provide recruitment flows (A separate rule is used if criteria are satisfied to provide overbank flows as opposed to recruitment flows). The targets for the entire recruitment hydrograph are set on the date to begin storage at Cochiti Lake for deviations, as input, or a computed date if no date was input. The computed date is based on an input number of days prior to the estimated time for the peak inflow to Cochiti. Deviations are not implemented if the current timestep is after the last year that deviations are implemented as input.

**Rule Execution:** The rule files if a switch is set to use an input date for Cochiti deviations as a value set to the UsePresetDayForCochitiDeviationsStorage that is greater than zero checked with the user-defined UsePresetDayToStartCochitiDeviationsStorage function and the current timestep is equal to the timestep to begin storage for deviations as input to the PresetDayForCochitiDeviationsStorage table slot in the MiddleValleyDemands data object or if the current timestep is equal to the computed date to begin storage for

deviations computed as the time of the peak inflow to Cochiti Lake offset by the number of days input to the DaysOfCochitiDeviationsStorageBeforePeak scalar slot in the MiddleValleyDemands data object and the current timestep is less than or equal to the value input to the LastYearCochitiDeviationsAuthorized table slot in the MiddleValleyDemands data object and the rule has not fired successfully for the current timestep as determined with reference to the predefined HasRuleFiredSuccessfully function.

*Rule Logic:* A FOREACH loop is used to reset targets at Central as set to the MinTargetForCentral series slot in the MiddleValleyDemands data object for the list of dates and associated targets created with the user-defined EnvironmentalMinRelease function in the RecruitmentAndOverbankFunctions utility group. This EnvironmentalMinRelease function identifies the list of targets to provide recruitment flows from the EnvironmentalTargets table slot in the MiddleValleyDemands data object and the associated dates for the assignment are set to begin four days before the estimated date for the peak inflow to Cochiti as determined with the user-defined CochitiMaxInflowDate function and continue until 25 days after the date of the peak. (Note that the targets may be reset back to the original table targets before the end of the 30-day hydrograph if storage for deviations at Cochiti Lake drops below a threshold volume before the end of the deviations period. Refer to the EndTargetsForOverbankOrRecruitment rule.)

Within the rule, targets are reset to provide recruitment flows if the Otowi forecast is within the range for conducting deviations to provide *recruitment* flows as checked with the user-defined IsRecruitmentForecastRange function and the projected peak inflow to Cochiti is within the range for conducting deviations to provide *recruitment* flows as input to the EnvironmentalMinMaxPeakFlows table slot in the MiddleValleyDemands data object or the Otowi forecast is within the range for conducting deviations to provide *overbank* flows but the projected peak inflow to Cochiti is less than the minimum for conducting deviations to provide *overbank* flows as input to the EnvironmentalMinMaxPeakFlows table slot.

An additional assignment statement is also included using the exact same logic to reset the targets for the SideCalcCentralTargetNoOtowiCutoff series slot in the MiddleValleyDemands data object. Cochiti deviations are not impacted by the policy for conserving supplemental leased San Juan-Chama Project water if a threshold year-to-date Otowi flow volume is exceeded, so the targets are recorded to this slot regardless of the year-to-date Otowi flow volume.

#### **A.8.16. ComputeCochitiRGConservationSpaceAvailable**

*Explanation:* This rule is used to set the annual value for conservation storage allowed at Cochiti Lake for Cochiti deviations if implemented. A storage amount is estimated based on the forecast volume and whether the operation is being conducted to provide

recruitment or overbank flows, and the allowed conservation storage is set on the same date that storage would begin for deviations.

*Rule Execution:* The rule files if a switch is set to use an input date for Cochiti deviations as a value set to the UsePresetDayForCochitiDeviationsStorage that is greater than zero checked with the user-defined UsePresetDayToStartCochitiDeviationsStorage function and the current timestep is equal to the timestep to begin storage for deviations as input to the PresetDayForCochitiDeviationsStorage table slot in the MiddleValleyDemands data object or if the current timestep is equal to the computed date to begin storage for deviations computed as the time of the peak inflow to Cochiti Lake offset by the number of days input to the DaysOfCochitiDeviationsStorageBeforePeak scalar slot in the MiddleValleyDemands data object and the current timestep is less than or equal to the value input to the LastYearCochitiDeviationsAuthorized table slot in the MiddleValleyDemands data object and the rule has not fired successfully for the current timestep as determined with reference to the predefined HasRuleFiredSuccessfully function.

*Rule Logic:* An IF THEN ELSE statement is used to set the value for the ComputedRGConservationSpaceAvailable annual series slot in the CochitiData data object depending on whether deviations are being implemented to provide recruitment flows or overbank flows.

If the Otowi forecast is within the range for conducting deviations to provide *recruitment* flows as checked with the user-defined IsRecruitmentForecastRange function and the projected peak inflow to Cochiti is within the range for conducting deviations to provide *recruitment* flows as input to the EnvironmentalMinMaxPeakFlows table slot in the MiddleValleyDemands data object or the Otowi forecast is within the range for conducting deviations to provide *overbank* flows but the projected peak inflow to Cochiti is less than the minimum for conducting deviations to provide *overbank* flows as input to the EnvironmentalMinMaxPeakFlows table slot, the allowable conservation space is set using the predefined TableLookup function with reference to the Otowi forecast as a percent of average and the *Recruitment Space* column in the RGConservationSpaceTable in the CochitiData data object.

If the Otowi forecast is within the range for conducting deviations to provide *overbank* flows as checked with the user-defined IsOverbankForecastRange function and the projected peak inflow to Cochiti is within the range for conducting deviations to provide *overbank* flows as input to the EnvironmentalMinMaxPeakFlows table slot in the MiddleValleyDemands data object or the Otowi forecast is within the range for conducting deviations to provide *recruitment* flows but the projected peak inflow to Cochiti is within the range for conducting deviations to provide *overbank* flows as input to the EnvironmentalMinMaxPeakFlows table slot, the allowable conservation space is set using the predefined TableLookup function with reference to the Otowi forecast as a percent of average and the *Overbank Space* column in the RGConservationSpaceTable in the CochitiData data object.

Otherwise, the value is set to the input value to the RGConservationSpaceAvailable table slot in the CochitiData data object.

### **A.8.17. EndTargetsForOverbankOrRecruitment**

*Explanation:* This rule is used to reset targets at Central set for Cochiti deviations back to the original table targets if conservation storage at Cochiti Lake drops below a threshold low volume. This adjustment is required to prevent supplemental leased San Juan-Chama Project water or Emergency Drought water from being used to provide recruitment or overbank flows after conservation storage is no longer available from Cochiti deviations.

*Rule Execution:* This rule executes if a value has been set in the MinTargetForCentral series slot in the Middle Valley Demands data object for the timestep equal to the current timestep plus one timestep (the approximate travel time from Abiquiu Dam to Cochiti) plus the travel time from Cochiti to Central and that value is greater than the value set with reference to the original target table and the storage in the RioGrandeConservation account at Cochiti Lake at the previous timestep is less than the input value in the LowStorageToEndDeviationsTargets scalar slot in the MiddleValleyDemands data object.

*Rule Logic:* A FOREACH loop is used to record the downstream flow target for the timestep equal to the current timestep plus one timestep (the approximate travel time from Abiquiu Dam to Cochiti) plus the travel time from Cochiti to Central for the MinTargetForCentral time series slot in the MiddleValleyDemands data object. If the year-to-date modeled flow volume in the Gage Outflow series slot for the Otowi stream gage object is less than the input value in the ThresholdOtowiVolumeHoldLeaseWater scalar slot in the MiddleValleyDemands data object or the storage in the SupplementalESA account at El Vado Reservoir at the previous timestep is greater than zero, the target is determined with reference to the user-defined MinTargetFlow function in the Get Data Functions utility group as a function of the date, hydrology year type, and target flow adjustment factor. The target flow, based on the date and hydrology year type, is input to the MinTargetFlows periodic slot in the MiddleValleyDemands data object, and the target flow adjustment factor for each target location and each month is input to the MinTargetFlowsSafetyFactor periodic slot in the MiddleValleyDemands data object. If the condition is not satisfied, the target is set to zero.

An outer IF THEN ELSE statement is included such that the year-to-date Otowi flow volume is not checked if the current timestep is the Start Timestep or January 1. An additional assignment statement is also included to record what the target would be if the values were not reset to zero after the year-to-date Otowi flow volume exceeds the threshold volume. This assignment is made to the SideCalcCentralTargetNoOtowiCutoff series slot in the Middle Valley Demands data object.

### **A.8.18. SetRioChamaDiversionAndDepletionRequested**

*Explanation:* This rule sets the diversion requested (and depletion requested) for the diversions from the Rio Chama below Abiquiu Dam based on input diversion values (and input fractional return flows). The diversions are adjusted based on input scale factors.

*Rule Execution:* If the current timestep is equal to the start timestep and the Diversion Requested time series slot for the Monastery diversion in the AbvAbiquiu aggregate diversion object is a NaN for the current timestep, the rule fires. The latter criterion ensures that the rule only fires once.

*Rule Logic:* A FOREACH loop is used to make the assignments for every timestep for the simulation period, and a second FOREACH loop is used to set the Diversion Requested and Depletion Requested for each diversion in the BelowAbiquiuDiversions subbasin (i.e. the diversions in the following five aggregate diversion objects: AbvAbiquiuDiversions, BlwAbiquiuDiversions, AbvConfluenceDiversions, BlwConfluenceDiversions, and BlwChamitaDiversions). The user-defined GetDiversionRequested function in the Scale Diversion Functions utility group is used to set the diversions based on input values in the RioChamaDiversionData data object and with consideration for input scale factors. The user-defined GetDepletionRequested function in the Scale Diversion Functions utility group is used to set the depletion requested based on the input diversion values and the input FractionalReturnFlows in the RioChamaDiversionData data object. Refer to Figure A.19 for a flowchart that depicts the logic used for this rule.

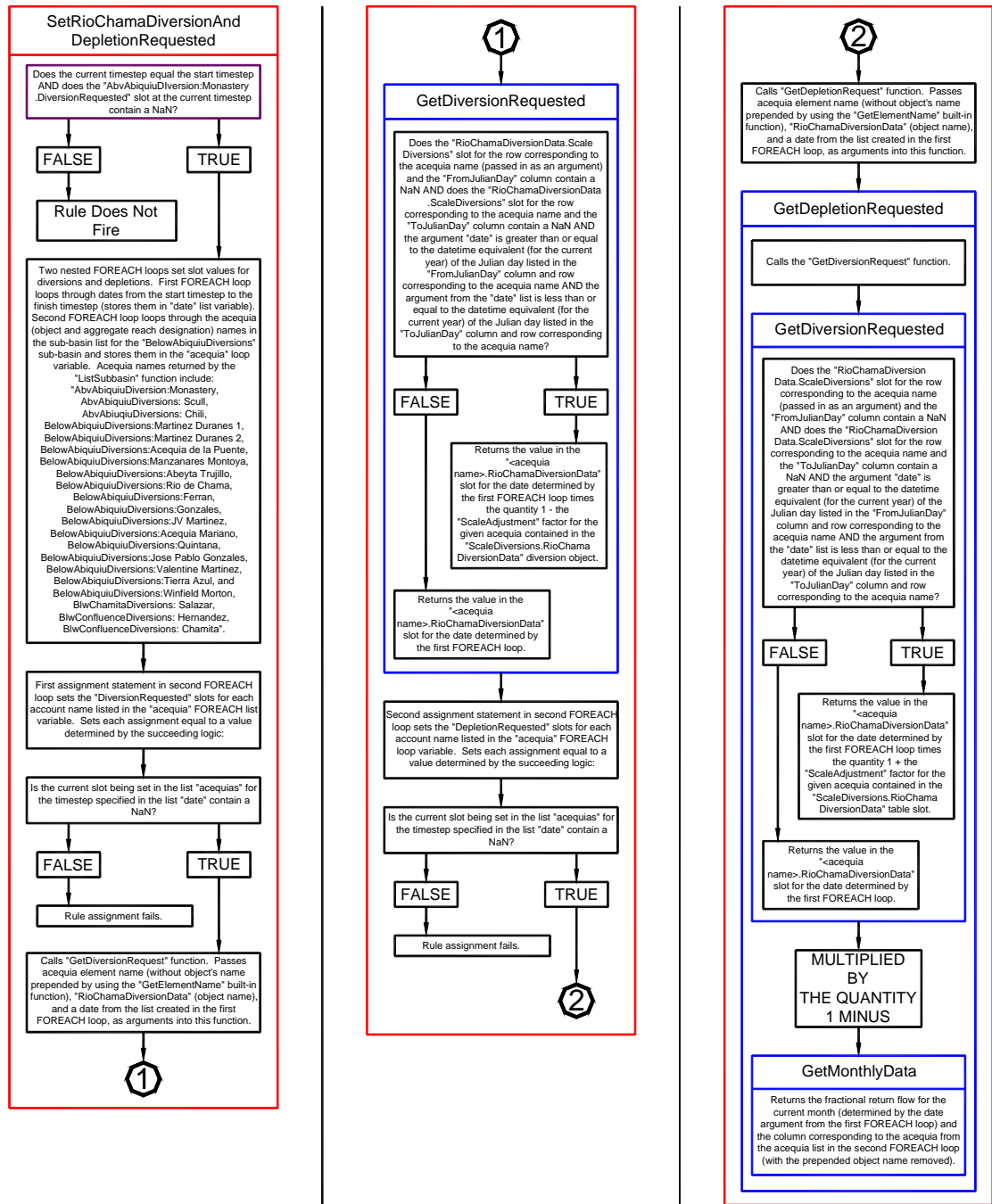


Figure A.19. Flowchart for the SetRioChamaDiversionAndDepletionRequested Rule



### **A.8.19. SetMiddleRioGrandeDiversionRequested**

*Explanation:* This rule sets the MRGCD diversion requested values for the separate water users on the aggregate diversion site objects for the diversions at Cochiti, Angostura, and Isleta based on input values and with consideration for input scale factors. (Diversions at San Acacia are set separately with the SanAcaciaLFCCDiversionsRequest and SanAcaciaSocorroMainCanalDiversionsRequest rules.)

*Rule Execution:* If it is the start timestep and the Diversion Requested time series slot for the EastSideMain diversion in the BlwCochitiDiversions aggregate diversion object is a NaN for the current timestep, the rule fires. The latter criterion ensures that the rule only fires once.

*Rule Logic:* A FOREACH loop is used to make the assignments for every timestep in the simulation, and a second FOREACH loop is used to set the Diversion Requested and Depletion Requested slots for each diversion in the MiddleRioGrandeDiversions subbasin. Refer to the following three aggregate diversion objects: BlwCochitiDiversions, BlwIsletaDiversions, and BlwSanFelipeDiversions (Diversions at the San Acacia diversion are set separately due to the dependency on the Unit 7 through flow). The user-defined GetDiversionRequested function in the Scale Diversion Functions utility group is used to set the diversions based on input values in the MiddleRioGrandeDiversionData data object and with consideration for input scale factors in that data object. IF THEN ELSE statements are included to decipher between elements in an aggregate diversion site object in the subbasin versus other objects that may be included in the subbasin.

### **A.8.20. ResetAngosturaDiversionForShortageOps**

*Explanation:* This rule increases the requested diversion values for the Angostura diversion when MRGCD is in a shortage situation (i.e. no water in storage) and operations are being conducted to assure delivery of Prior and Paramount water to the six Middle Valley pueblos. In actual operations, diversions are also increased so MRGCD can then use the limited supply as efficiently as possible.

*Rule Execution:* The rule fires if the rule has not fired successfully for the current timestep as determined with reference to the predefined HasRuleFiredSuccessfully function and the current timestep is greater than or equal to March 1<sup>st</sup> and the current timestep plus one timestep (the approximate travel time from Abiquiu Dam to Cochiti) plus the travel time from Cochiti to San Marcial is less than or equal to October 31<sup>st</sup> (i.e. the diversion is being reset only within the irrigation season) and the MRGCD supply as computed with the user-defined MRGCDSupply function is less than the current MRGCD daily demand.

*Rule Logic:* A FOREACH loop is used to reset the diversion requested values from the current timestep through the current timestep plus one timestep (the approximate travel time from Abiquiu Dam to Cochiti) plus the travel time from Cochiti to San Marcial. Values are reset for the AlbMainCanal and AtriscoFeeder water users on the AngosturaDiversion aggregate diversion site object to the maximum canal capacities as input to the MaxCapacityAtAngosturaAlbMainCanal and MaxCapacityAtAngosturaAtriscoFeeder scalar slots on the MiddleRioGrandeDiversionData data object.

### A.8.21. SetPreemptiveAlbuquerqueCutoffSwitch

*Explanation:* This rule sets a switch to identify whether conditions are satisfied for a preemptive cutoff of Albuquerque surface water diversions (Albuquerque would switch to groundwater to meet their demand). Preemptive cutoff criteria include a low river flow at which Albuquerque would shutdown before the permit criteria result in a curtailment to diversions, a high Cochiti outflow at which operating the diversion would be unsafe and impractical, and an Abiquiu high outflow at which Abiquiu operations are being conducted for flood control operations and Albuquerque's San Juan-Chama Project water would not be released. Refer to Figure A.20 for a flowchart that depicts the preemptive cutoff criteria with current input values.

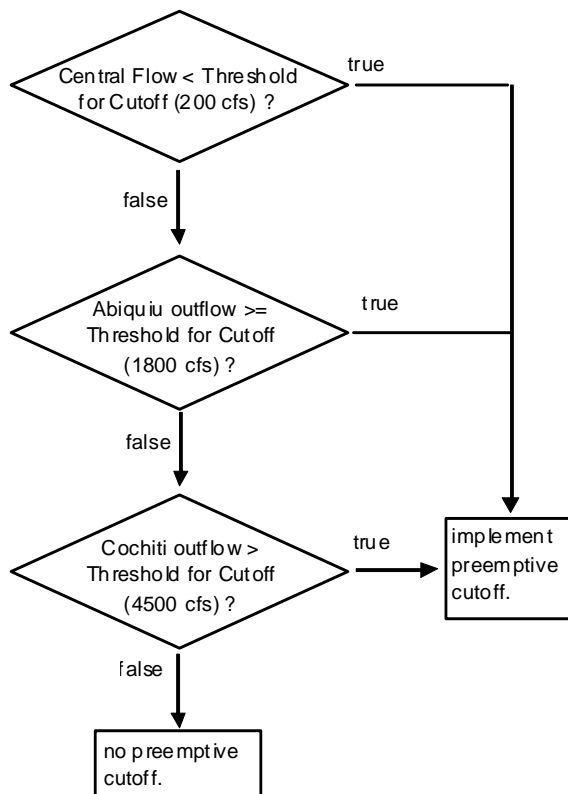


Figure A.20. Criteria Checked for a Preemptive Cutoff of Albuquerque Diversions

*Rule Execution:* This rule fires every timestep.

*Rule Logic:* An IF THEN ELSE statement is used to set the value for the PreemptiveCutoffSwitch series slot for the current timestep in the MiddleValleyDemands to 1.0 if the criteria in the PreemptiveCutoffAlbuquerqueDiversion function are satisfied. Otherwise, the value is set to 0.0.

The PreemptiveCutoffAlbuquerqueDiversion function includes a check to see if the Gage Outflow on the Central stream gage object at the previous timestep is less than the input value for the ThresholdCentralFlowForAlbPreemptiveCutoff scalar slot in the MiddleValleyDemands data object or the Outflow from the Abiquiu level power reservoir object is greater than or equal to the input value in the ThresholdHighAbiquiuOutflowForAlbPreemptiveCutoff scalar slot in the MiddleValleyDemands data object or the Outflow from the Cochiti Lake object is greater than the input to the ThresholdHighCochitiOutflowForAlbPreemptiveCutoff scalar slot on the MiddleValleyDemands data object.

## **A.8.22. SetAlbuquerqueDiversion**

*Explanation:* This rule is used to set the Albuquerque diversion. The diversion is set into the future as needed for the hypothetical simulation to compute the flow needed for targets. If preemptive cutoff criteria are not satisfied, the Albuquerque diversion is set to twice the input maximum demand if permit restrictions for curtailment or cutoff do not apply, but the preemptive cutoff criteria would cut off the diversions before the permit based on the current model inputs. Diversions are set to twice the input demand under the assumption that half of the diversion will be returned to the river. Albuquerque will divert twice the amount of their San Juan-Chama Project water that is delivered. Diversions are not restarted until preemptive cutoff criteria have not been satisfied for at least two weeks and the flow at Central has exceeded an input threshold for restarting diversions.

*Rule Execution:* If the rule has not fired successfully for the current timestep as determined with reference to the predefined HasRuleFiredSuccessfully function, the rule fires.

*Rule Logic:* A FOREACH loop is used to set the Diversion Requested series slot on the AlbuquerqueWaterUser water user object for the current timestep through the current timestep plus one timestep (the approximate travel time from Abiquiu Dam to Cochiti) plus the travel time from Cochiti to San Marcial. The diversion is set to zero if the current timestep is equal to the Start Timestep or the current year is prior to the start year for diversions as input to the AlbuquerqueStartYear table slot on the MiddleRioGrandeDiversionData data object referenced using the user-defined AlbuquerqueStartYear function in the Get Data Functions utility group or the criteria for a preemptive cutoff are satisfied for the current timestep as checked with the user-defined PreemptiveCutoffAlbuquerqueDiversion function or the preemptive cutoff criteria were

satisfied within the last two weeks of simulation as checked with the user-defined `PreemptiveCutoffAlbuquerqueDiversionsWithinLastTwoWeeks` function or the `Diversion Requested` value was zero on the previous timestep and the value for the Gage Outflow series slot on the Central stream gage object at the previous timestep is less than the value input to the `ThresholdCentralFlowForAlbRestart` scalar slot on the `MiddleValleyDemands` data object. Otherwise, the diversion is set using the user-defined `AlbuquerqueDiversion` function in the Demand Functions utility group. Refer to Figure A.21 for a flowchart that shows the criteria checked daily when setting the Albuquerque diversions.

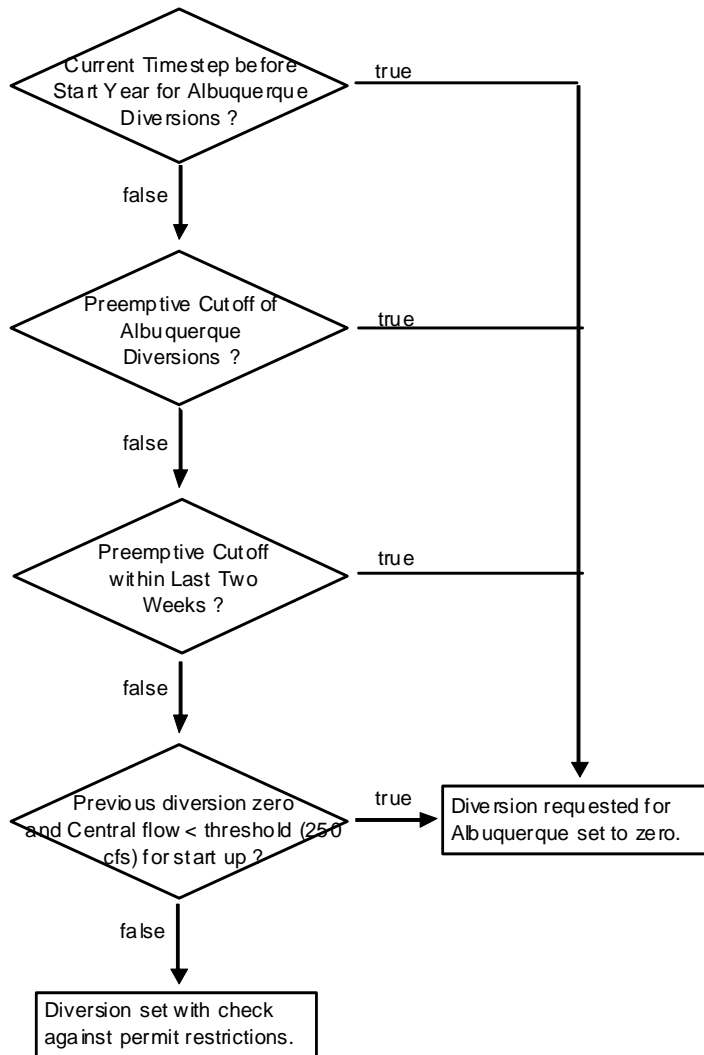


Figure A.21. Criteria Checked Daily when Setting the Albuquerque Diversion Requested

The `AlbuquerqueDiversion` function is used to set the diversion with appropriate consideration for the permit criteria for curtailment and cutoff thresholds as input to the `AlbuquerqueCurtailmentRanges` table slot in the `MiddleRioGrandeDiversionData` data object. There are four IF THEN ELSE statements in the `AlbuquerqueDiversion` function. For the outside IF THEN ELSE statement, if the river flow (`AlamedaBridge.Gage`

Outflow slot) is less than the curtailment threshold but greater than the cutoff threshold, the diversion is set to twice the input diversion as input to the Albuquerque time series slot in the MiddleValleyDemands data object minus the difference between the threshold curtailment flow and the river flow IF Albuquerque also has more than that amount of water in its storage account at Abiquiu. For the second IF THEN ELSE statement, if Albuquerque does not have that amount of water in its storage account at Abiquiu OR if Albuquerque does not have the volumetric amount of the input diversion in its storage account at Abiquiu, the diversion is set to twice the amount of water in its storage account. For the third IF THEN ELSE statement, if the flow in the river is less than the cutoff threshold or Albuquerque does not have any water in its storage account at Abiquiu, the diversion is set to zero. For the final IF THEN ELSE statement (that is, conditions when the river flow is greater than the curtailment flow), if the amount of water that Albuquerque has in its storage account at Abiquiu is less than the input diversion demand, the diversion is set to twice the amount of water in its storage account at Abiquiu. Else, the diversion is set to twice the input demand. A constraint is placed on this function to assure the result is greater than 0.0 cfs. Refer to Figure A.22 for a flowchart that depicts the permit criteria for curtailment and cutoff per the permit with reference to the current input value of 65 cfs for their demand.

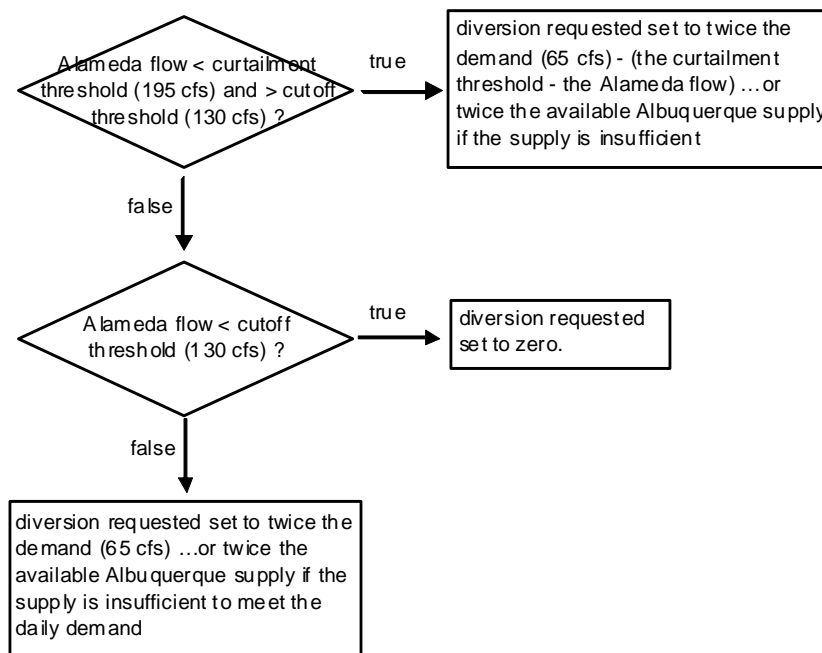


Figure A.22. Permit Restrictions Checked when Setting Albuquerque Diversions

A second assignment statement is included in the rule within the FOREACH loop to also set the value for the Depletion Requested series slot on the AlbuquerqueWaterUser water user object using the exact same calculation.

### **A.8.23. CheckPreemptiveCutoffVersusPermitTEST**

*Explanation:* This rule is used to check what Albuquerque diversions would be solely by checking permit conditions as a side calculation.

*Rule Execution:* This rule fires every timestep.

*Rule Logic:* A FOREACH loop is used to set the values for the SideCalcAlbPermitDiversion series slot in the MiddleValleyDemands data object to the result from the user-defined AlbuquerqueDiversion function that checks permit criteria for setting the Albuquerque diversions.

### **A.8.24. InitialEstimateElVadoAndAbiquiuAndCochitiInflow**

*Explanation:* An estimated inflow to Cochiti Lake is used to evaluate the available flow for MRGCD if no additional water is available in storage for MRGCD. That inflow is then referenced when potentially setting shorted diversion requested values at the MRGCD diversions to assure supplemental water for targets is not diverted when a full supply would otherwise not be available for MRGCD.

The estimated inflow to Cochiti Lake is computed with separate steps to first estimate the inflow of native Rio Grande water to El Vado and the subsequent inflow to Abiquiu Reservoir. This rule is used to compute initial estimates for the inflow to each reservoir.

The initial estimate for the inflow to El Vado Reservoir is based on the input local inflow above El Vado and the input flow of native Rio Grande water from Willow Creek. A percent forecast error is applied if it is not a NaN. The initial estimate for the inflow of Rio Grande water to Abiquiu Reservoir is computed by subtracting the inflow of San Juan-Chama Project water with consideration for the loss rate for San Juan-Chama Project water between El Vado Dam and Abiquiu Reservoir and subtracting the amount of Rio Grande water that may have been transferred to Reclamation's storage account at Abiquiu Reservoir. The initial estimate for the inflow of Rio Grande water to Cochiti Lake is computed by subtracting the inflow of San Juan-Chama Project water, with consideration for the loss rate for San Juan-Chama Project water between Abiquiu Dam and Cochiti Lake, from the total inflow to Cochiti Lake.

*Rule Execution:* If the current timestep is greater than the start timestep, the rule fires.

*Rule Logic:* The ForecastElVadoInflow function in the Estimated Inflow Functions utility group includes an IF THEN ELSE statement to apply the PercentForecastError for the month as recorded in the ForecastData data object if it is not a NaN. The user-defined EstimateElVadoInflow function in the Estimate Inflow Functions utility group is used to estimate the inflow to El Vado Reservoir. This function sums the inflow from Willow Creek for the previous timestep as input to the Inflow2 time series slot in the AzoteaWillow confluence object and the input Local Inflow in the ElVadoLocalInflow

reach object for the previous timestep. Those flows are converted to volumes using the predefined SumFlowsToVolume function. Within the rule, the result is converted back to a flow using the predefined VolumeToFlow function. The result is recorded to the EstimatedInflow time series slot for the previous timestep in the ElVadoData data object.

The user-defined PreviousRGInflow function in the General Compute Functions utility group can be used to compute the inflow of Rio Grande water for the previous timestep to El Vado, Abiquiu, or Cochiti. The function is used in this rule for Abiquiu and Cochiti Lakes. For Abiquiu Reservoir, the RioGrandeAbiquiuToReclamationAbiquiu supply for the previous timestep and all the San Juan supplies from El Vado for the timestep before the previous timestep are subtracted from the value of the Inflow time series slot in the Abiquiu level power reservoir object for the previous timestep. To compute the sum of the San Juan supplies from El Vado, a list of the supplies is developed with the user-defined AllSJSuppliesFrom function in the Account Lists utility group, and the supplies are summed using the user-defined SumSuppliesAtDate function in the Account Summing Functions utility group. The loss rate for San Juan water from El Vado Dam to Abiquiu Reservoir is applied using the value in the Losses table slot in the SanJuanChamaRules data object which is referenced using the user-defined SJCLoss function in the Get Data Functions utility group. The result from the rule is recorded to the EstimatedAbiquiuInflow time series slot in the AbiquiuData data object for the previous timestep.

For Cochiti Lake, all the San Juan supplies to Cochiti Lake are identified using the user-defined AllSuppliesByReleaseTypeDestinations function in the Account Lists utility group and summed using the user-defined SumSuppliesAtDate function in the Account Summing Functions utility group for the timestep before the previous timestep. The loss rate for San Juan water from Abiquiu Dam to Cochiti Lake is applied using the value input to the Losses table slot in the SanJuanChamaRules data object which is referenced with the user-defined SJCLoss function in the Get Data Functions utility group. That result is subtracted from the value for the Inflow time series slot in the Cochiti storage reservoir object for the previous timestep to get the estimated inflow of Rio Grande water that is recorded to the EstimatedInflow time series slot in the CochitiData data object for the previous timestep. Refer to Figure A.23 for a flowchart depicting the logic used for the three assignment statements in this rule.

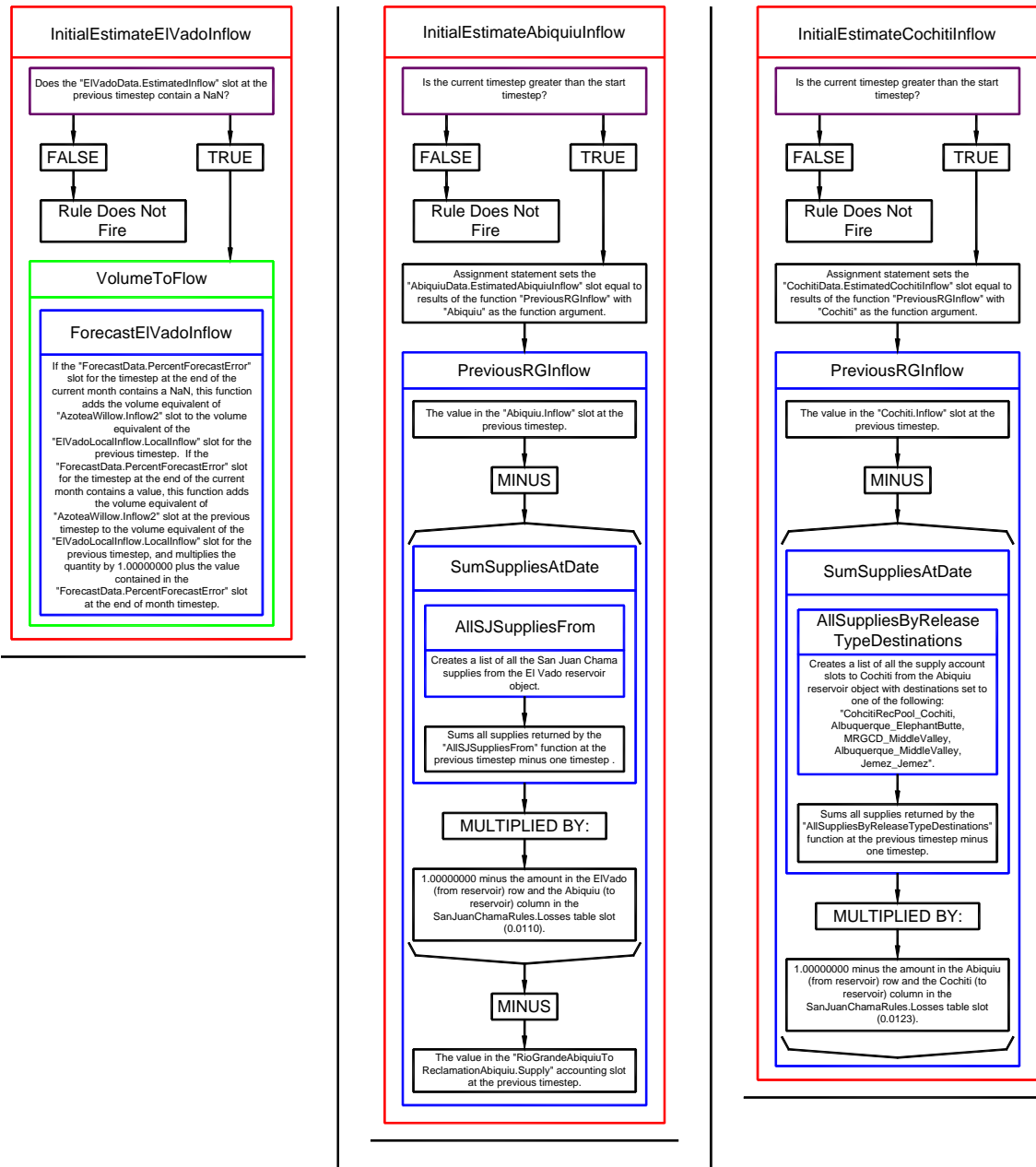


Figure A.23. Flowchart for the InitialEstimateElVadoAndAbiquiuAndCochitiInflow Rule

### A.8.25. Set Estimated ElVado Inflow

*Explanation:* Initial estimates for the inflow to El Vado Reservoir for the current timestep plus the next nine timesteps are computed based on the local inflow above El Vado Reservoir and the flow from Willow Creek. The result is needed for ultimately developing an estimated inflow to Cochiti Lake referenced when setting shorted MRGCD diversions when MRGCD has no more water in storage. Values are needed for a few



timesteps out from the current timestep because of the travel time to the MRGCD diversions.

*Rule Execution:* If the rule has not fired successfully for the current timestep as determined with reference to the predefined HasRuleFiredSuccessfully function, the rule fires.

*Rule Logic:* A FOREACH loop is used to set the values for the EstimatedInflow time series slot for the current timestep plus the next nine timesteps. The user-defined EstimateElVadoInflowList function in the Estimate Inflow Functions utility group is used to estimate the inflows to El Vado Reservoir. This function sums the inflow from Willow Creek as input to the Inflow2 time series slot in the AzoteaWillow confluence object with the input Local Inflow in the ElVadoLocalInflow reach object for the corresponding date referenced in the FOREACH loop. The results are recorded to the EstimatedInflow time series slot in the ElVadoData data object.

#### **A.8.26. Set Estimated Abiquiu Inflow**

*Explanation:* Initial estimates for the inflow to Abiquiu Reservoir for the next eight timesteps are computed. If the stipulations of Article VII of the Compact are in effect, all the estimated inflow to El Vado Reservoir is counted (i.e. it is assumed El Vado inflows are bypassed). Otherwise, a percentage of the estimated inflow to El Vado Reservoir is considered based on the percentage of inflow bypassed at the previous timestep. So, it is assumed the same approach used for filling El Vado Reservoir at the previous timestep will continue to apply. The resulting inflow to Abiquiu Reservoir is needed for ultimately developing an estimated inflow to Cochiti Lake referenced when setting shorted MRGCD diversions when MRGCD has no more water in storage.

*Rule Execution:* If the rule has not fired successfully for the current timestep as determined with reference to the predefined HasRuleFiredSuccessfully function, the rule fires.

*Rule Logic:* The user-defined EstimateAbiquiuInflowList function in the Estimate Inflow Functions utility group is used to estimate the inflows to Abiquiu Reservoir. The predefined HypSim function is used to determine resulting values for the Outflow time series slot in the ElVadoToAbiquiuLocalInflow reach object (linked to the Inflow time series slot in the Abiquiu level power reservoir object) using the objects in the BelowElVado subbasin and setting the values for the Gage Inflow time series slot in the BlwElVado stream gage object equal to the EstimatedInflow time series slot in the ElVadoData data object. A FOR DO loop is then used with an APPEND ONTO statement to create a list of the results and the corresponding date for the next eight timesteps. The resulting list of values and dates are referenced in the rule to set the values for the EstimatedAbiquiuInflow time series slot for the next eight timesteps using a FOREACH loop.

### **A.8.27. Set Estimated Cochiti Inflow**

*Explanation:* Initial estimates for the inflow to Cochiti Lake for the next seven timesteps are computed assuming estimated inflows to Abiquiu Reservoir are bypassed and adding any letter water deliveries below Abiquiu. The results are used for estimating shorted MRGCD diversions when MRGCD has no water in storage.

*Rule Execution:* If the value of the EstimatedAbiquiuInflow time series slot in the AbiquiuData data object for the current timestep is not a NaN and if the rule has not fired successfully for the current timestep as determined with reference to the predefined HasRuleFiredSuccessfully function, the rule fires.

*Rule Logic:* The user-defined EstimateCochitiInflowList function in the Estimate Inflow Functions utility group is used to estimate the inflows to Cochiti Lake for the next timestep through the current timestep plus seven days. Within a WITH DO loop, the predefined HypSim function is used to determine resulting values for the Outflow time series slot in the OtowiToCochitiLocalInflow reach object (linked to the Inflow time series slot in the Cochiti storage reservoir object) using the objects in the BelowAbiquiu subbasin and if the values for the Gage Inflow time series slot in the BlwAbiquiu stream gage object are set equal to the values in the EstimatedAbiquiuInflow time series slot in the AbiquiuData data object plus letter water deliveries from Abiquiu as determined with the user-defined EstimateLetterWaterBelowAbiquiu function. After the HypSim function within the EstimateCochitiInflowList function, a WITH DO loop is used with a FOR DO loop and an APPEND ONTO statement to create a list of the results for the next seven timesteps and the corresponding date. The resulting list of values and dates are referenced in the rule to set the values for the EstimatedInflow time series slot for the next seven timesteps using a FOREACH loop.

### **A.8.28. InitialFlowToMeetCentralAndIsletaAndSanAcaciaAndSanMarcial**

*Explanation:* This rule sets values for the minimum release to meet the target flow at Central, Isleta, San Acacia, and San Marcial to 0.0 cfs for the Start Timestep. This rule is required because the flow needed for each target in the Middle Valley must be set for the next timestep, due to the travel time from Abiquiu Dam to Cochiti Lake, such that the flow needed at Abiquiu Dam for the current timestep can then be determined. Based on the approach, values must be set separately at the Start Timestep.

*Rule Execution:* If the value for the MinReleaseForCentral time series slot in the MiddleValleyDemands data object for the current timestep is a NaN and it is the start timestep, the rule fires. The former check ensures that the rule only fires once.

*Rule Logic:* Four assignment statements are used to set the MinReleaseForCentral, MinReleaseForIsleta, MinReleaseForSanAcacia, and MinReleaseForSanMarcial time series slots in the MiddleValleyDemands data object for the current timestep 0.0 cfs.

### A.8.29. Shorted Middle Valley Diversions

*Explanation:* This rule is used to reset the MRGCD requested diversions to lower amounts if the diversions would be shorted due to a limited supply. This adjustment then prevents supplemental water used for meeting targets from being diverted by MRGCD. Shorted diversions are estimated based on the estimated inflow to Cochiti as determined with the Set Estimated Cochiti Inflow rule if the estimated inflow is less than the current daily MRGCD demand at Cochiti. Note that requested diversions at a particular diversion are NOT adjusted if there are no downstream targets (i.e. Isleta diversions will not be shorted if there is a Central target but not targets below Isleta).

*Rule Execution:* This rule fires if the rule has not fired successfully for the current timestep as determined with reference to the predefined HasRuleFiredSuccessfully function and the result from the user-defined AnyMinTargetIsGreaterThanZero function is true and the value for the EstimatedInflow series slot in the CochitiData data object is not a NaN and supplemental water is available to meet targets checked if the sum of the storage in the Reclamation account at Heron, the Reclamation and Supplemental ESA accounts at El Vado, the Reclamation and Rio Grande Conservation accounts at Abiquiu and the Rio Grande Conservation account at Cochiti is greater than zero and the inflow to Cochiti available for MRGCD is not very large checked if the EstimatedInflow to Cochiti is less than 2.5 times the value in the MRGCD time series slot in the MiddleValleyDemands data object for the current timestep or 2.0 times the value for the MinTargetForCentral time series slot in the Middle Valley Demands data object for the current timestep. The latter two checks are included to minimize the amount of hypothetical simulation invoked during a simulation with this rule.

*Rule Logic:* A FOREACH loop is used in the rule to set the diversion values for the diversion objects, aggregate diversion site objects, or water user objects in the DiversionsCochitiToSanMarcialFloodway subbasin for the current timestep plus the next seven timesteps. The user-defined ShortedMiddleValleyDiversions function in the Silvery Minnow Functions utility group is used to create a multi-dimensional list with the name of the slot to be set, the value for the diversion requested for that slot, and the corresponding date. An IF THEN statement is used to only reset the diversion requested values if there are targets downstream of that diversion. The GET SLOT @INDEX, GET DATETIME @INDEX, and GET NUMERIC @INDEX statements are used in the rule to reference the data in the array. This rule also includes an assignment statement to set the value of the ShortedMiddleValleyDiversions time series slot in the RuleHasFired data object to 1.0 for the current timestep.

In the ShortedMiddleValleyDiversions function, the diversion requested is determined depending on whether the diversion is set up as a diversion object, aggregate diversion site object, or water user object. Several different user-defined functions in the General List Functions and Account Lists utility groups are used to create lists of the diversions by diversion type. The user-defined MiddleValleyHypSim function in the

HypotheticalSimulation Functions utility group is used to determine, based on a hypothetical simulation of all the objects in the BlwCochitiToSanMarcial subbasin, values for the Total Diversion or Diversion time series slots for each type of diversion. The hypothetical simulation is completed assuming values for the Inflow2 time series slot in the RioGrandeJemez confluence object are equal to the inflow to Jemez Lake, and the estimated flow below Cochiti Dam is also known based on the result from the user-defined CochitiReleaseShortingMiddleValleyDiversions function in the HypotheticalSimulation Functions utility group (Refer to Section A.8.29.1 for further discussion of this function). IF THEN ELSE statements are then used within a FOREACH statement to set the diversion requested depending on the type of object in the model for the different dates.

#### A.8.29.1. CochitiReleaseShortingMiddleValleyDiversions

*Explanation:* This function is used to determine an estimated flow below Cochiti Dam for the hypothetical simulation in the MiddleValleyHypSim function. The release to MRGCD is assumed to equal the estimated inflow if the inflow is greater than the MRGCD Demand. Otherwise, the release to MRGCD is assumed to match the MRGCD Demand if MRGCD has enough water available in storage to make up for the difference between the MRGCD demand and the estimated inflow. If MRGCD does not have enough water in storage, the release is assumed to equal the greater of the available water in storage for MRGCD, the P&P demand (not to exceed the available P&P water in storage), or the estimated inflow. Any release from storage for the Albuquerque surface water diversions is added to the determined flow available for MRGCD since Albuquerque diversions are represented as part of the hypothetical simulation of the Middle Valley.

*Rule Logic:* This function estimates a value for the flow below Cochiti Dam based on the inflow to Cochiti Lake, as referenced using the user-defined CochitiEstimatedFlow function in the HypotheticalSimulation Functions utility group and input values to the MRGCD and Indian Demand time series slots in the MiddleValleyDemands data object as referenced with the user-defined MRGCDDemandAtGivenDate and IndianDemandAtGivenDate functions, respectively, in the Get Data Functions utility group. If the estimated inflow to Cochiti Lake is greater than the MRGCD demand, the outflow is estimated to be the same as the inflow plus any release of San Juan-Chama Project water for the Albuquerque diversion. Otherwise, if the estimated inflow to Cochiti Lake is less than the MRGCD demand and MRGCD also has enough Rio Grande and/or San Juan-Chama Project water available in storage at El Vado, Abiquiu, and Cochiti Lake, as determined with the user-defined SumAllAvailableMRGCDStorageAvailableToMeetMRGCDDemand function in the General Account Functions utility group, to meet its demand, the outflow is assumed to equal the MRGCD demand plus any release of San Juan-Chama Project water for the Albuquerque diversion. Otherwise, the outflow is assumed to equal the maximum of the amount of water available in storage for MRGCD as determined with the EstimateMRGCDStorageAvailableToMeetMRGCDDemand function, the magnitude of the Indian Demand (or the storage in the Indian storage account at El Vado Reservoir, as

determined with the user-defined `AvailableIndianStorageAvailableToMeetIndianDemand` function in the General Account Functions utility group) or the estimated Cochiti inflow plus any release of San Juan-Chama Project water for the Albuquerque diversion.

### **A.8.30. ComputeReleaseToMeetMinimumCentralFlow**

*Explanation:* This rule uses hypothetical simulation to determine the minimum total outflow from Cochiti Dam needed to meet the determined target flow at Central. The value for the next timestep is computed as needed to eventually determine the flow needed at Abiquiu at the current timestep. Middle Valley tributary inflows are assumed to equal zero while inflows to Jemez Reservoir are assumed to be bypassed at Jemez Dam.

*Rule Execution:* The rule fires if the recorded value to the `MinTargetForCentral` series slot in the `MiddleValleyDemands` data object for the current timestep is greater than zero and the rule has not fired as determined with a reference to the predefined `HasRuleFiredSuccessfully` function and if the sum of the storage in the Reclamation account at Heron, the Reclamation and Supplemental ESA accounts at El Vado, the Reclamation and Rio Grande Conservation accounts at Abiquiu and the Rio Grande Conservation account at Cochiti is greater than zero. Also, the `EstimatedInflow` to Cochiti is checked to assure it is less than 2.5 times the value in the `MRGCD` time series slot in the `MiddleValleyDemands` data object for the current timestep or 2.0 times the value for the `MinTargetForCentral` time series slot in the Middle Valley Demands data object for the current timestep. The checks are included to minimize the amount of hypothetical simulation started with this rule.

*Rule Logic:* The value for the `MinReleaseForCentral` time series slot in the `MiddleValleyDemands` data object for the next timestep is set using the user-defined `ReleaseToMeetMinFlowAtCentral` function in the `HypotheticalSimulation Functions` utility group. This function uses the predefined `HypTargetSimWithStatus` function to determine the required outflow from Cochiti (or the value for the `Inflow` time series slot for the `BlwCochitiDiversionReach` reach object) to meet a target flow at Central (or the `Gage Outflow` time series slot in the Central stream gage object) at the timestep equal to the next timestep plus the number of timesteps for the release to reach Central from Cochiti Dam. The next timestep and travel time are set as function arguments within the rule where the travel time is input to the `ApproxNoOfDaysDS` table slot in the `CochitiData` data object and referenced with the user-defined `ApproximateNumDaysDownstream` function in the `Get Data Functions` utility group.

The function completes the hypothetical simulation for all the objects in the `BlwCochitiToCentralTarget` subbasin and uses the minimum outflow and maximum outflow from Cochiti Dam as the bounds for the solution procedure as input to the `MinRelease` periodic slot and `ReservoirData` table slot, respectively, in the `CochitiData` data object and referenced with the user-defined `MinOutflow` and `MaxOutflow` functions, respectively, in the `Get Data Functions` utility group. A tolerance of 0.25 cfs is allowed

for the target flow. The inflow from the Rio Jemez is set to the inflow to Jemez Reservoir with the user-defined JemezInflowInputs function and tributary inflows are set to zero using the user-defined TributaryInflowInputs function.

#### **A.8.31. ComputeReleaseToMeetMinimumIsletaFlow**

*Explanation:* This rule is essentially the same as the ComputeReleaseToMeetMinimumCentralFlow Rule except the minimum release is being determined for the target flow at Isleta (Refer to Section A.8.30 for more details).

#### **A.8.32. ComputeReleaseToMeetMinimumSanAcaciaFlow**

*Explanation:* This rule is essentially the same as the ComputeReleaseToMeetMinimumCentralFlow Rule except the minimum release is being determined for the target flow at San Acacia (Refer to Section A.8.30 for more details).

#### **A.8.33. ComputeReleaseToMeetMinimumSanMarcialFlow**

*Explanation:* This rule is essentially the same as the ComputeReleaseToMeetMinimumCentralFlow Rule except the minimum release is being determined for the target flow at San Acacia (Refer to Section A.8.30 for more details).

#### **A.8.34. SetCochitiMinimumFlow**

*Explanation:* This rule determines the maximum value from the four computed minimum releases from Cochiti Dam to meet the target flows at Central, Isleta, San Acacia, and San Marcial. The resulting maximum total flow needed for targets in the Middle Valley is then referenced later to identify the amount of supplemental water needed from Abiquiu Reservoir. Values are set for both the current timestep and the next timestep. (Due to the varying travel time from Abiquiu Dam to Cochiti Lake, flows needed for the current timestep and next timestep had *previously* been averaged when determining the flow needed from Abiquiu Reservoir. This approach is not used in the current ruleset, but the minimum flow needed at Cochiti Lake is still set for both the current timestep and next timestep.)

*Rule Execution:* If the value for the MinimumFlow time series slot in the MiddleValleyDemands data object is a NaN for the next timestep, the rule fires.

*Rule Logic:* A FOREACH loop is used to set the value for the MinimumFlow time series slot in the MiddleValleyDemands data object, if it is a NaN, for the current timestep and the next timestep with the user-defined ComputeCochitiMinimumFlow function in the Cochiti Account Functions utility group. This function uses the predefined MaxItem

function to determine the maximum value from the minimum releases to meet the target flows at Central, Isleta, San Acacia, and San Marcial.

### **A.8.35. ComputeAbiquiuMRGCDDemand**

*Explanation:* This rule uses hypothetical simulation to determine the required release from Abiquiu Dam to meet the MRGCD demand at Cochiti Lake. A computed value is determined first and then a final value is set with the SetAbiquiuMRGCDDemand rule. The separate steps are included to potentially allow for some alternate approaches to be used in regards to better matching the travel time for needed deliveries. This value is then referenced later to determine the release needed from El Vado Reservoir and to ultimately set any potential releases from storage to meet the MRGCD demand.

*Rule Execution:* If the value for the ComputedMRGCDDemand time series slot in the AbiquiuData data object is a NaN for the current timestep and the value for the MRGCDDemand in the AbiquiuData data object is a NaN for the next timestep, the rule fires.

*Rule Logic:* The value for the ComputedMRGCDDemand time series slot in the AbiquiuData data object for the current timestep is set using the user-defined AbiquiuComputedMRGCDDemand function in the Demand Functions utility group. This function uses the predefined HypTargetSimWithStatus function to determine the required outflow from Abiquiu Dam (or the value for the Gage Inflow time series slot for the BlwAbiquiu stream gage object) to meet a target inflow to Cochiti Lake (or a value for the Outflow time series slot in the OtowiToCochitiLocalInflow reach object) equal to the value for the MRGCD time series slot in the MiddleValleyDemands data object for the next timestep. The function completes the hypothetical simulation for all the objects in the BelowAbiquiu subbasin and uses the input minimum and maximum outflows from Abiquiu Dam of 0 and 1800 cfs, respectively (the channel capacity below Abiquiu Dam is 1800 cfs) as the bounds for the solution procedure. A tolerance of 2 cfs is allowed for the target flow.

### **A.8.36. SetAbiquiuMRGCDDemand**

*Explanation:* This rule sets the MRGCD demand for the next timestep based on the computed demand for the current timestep. This separate rule allows for an adjustment to the computed demand to be implemented to potentially better match the varying travel time from Abiquiu Dam to Cochiti Lake.

*Rule Execution:* If the value for the MRGCDDemand in the AbiquiuData data object is a NaN for the next timestep, the rule fires.

*Rule Logic:* A FOREACH loop is used to set the value for the MRGCDDemand time series slot in the AbiquiuData data object for the current timestep and the next timestep

equal to the value for the ComputedMRGCDDemand time series slot in the AbiquiuData data object for the current timestep.

### **A.8.37. SetRaftingRelease**

*Explanation:* This rule is turned off, but it records a value for the rafting release from El Vado Dam. The logic in this rule needs to be reviewed and checked.

*Rule Execution:* If the value for the ComputedRaftingSchedule time series slot in the ElVadoData data object for the current timestep is a NaN, the rule fires.

*Rule Logic:* The value for the ComputedRaftingSchedule time series slot in the ElVadoData data object for the current timestep is set with the user-defined ElVadoRaftingRelease function in the Rafting Release Functions utility group. This function includes a check to see if it is the rafting season as determined by the user-defined ElVadoRaftingSeason function in the Rafting Release Functions utility group. This check is based on values input to the RaftingSeasonDates table slot in the ElVadoData data object. These values are referenced using the user-defined RaftingSeasonStartDate and RaftingSeasonEndDate functions in the Get Data Functions utility group which make the necessary adjustments for leap years. The rafting season starts on the first Friday after the input start date, so in the ElVadoRaftingSeason function, depending on the day of the week, adjustments are made to assure the start day is after the first Friday after the start date. This is done with a series of seven IF THEN ELSE statements. If it is not the rafting season, the recorded release is 0.0 cfs.

If it is the rafting release season, the release is computed as the input rafting release for the day of the week minus the predetermined outflow from El Vado Dam, but the release is also restricted based on the amount of water MRGCD has in its storage account at El Vado Reservoir that can be moved to its storage account at Abiquiu Reservoir. For the former, the input rafting release minus the current outflow is calculated using the user-defined PreliminaryRaftingRelease function in the ElVado Functions utility group. The input rafting release values are in the RaftingSchedule table slot in the ElVadoData data object. The result from the PreliminaryRaftingRelease function is restricted to be greater than or equal to 0.0 cfs. The latter is computed as the amount of water in storage accounts at El Vado Reservoir for accounts that also have storage accounts at Abiquiu Reservoir minus the amount of that water to be used to payback debts to Otowi or Albuquerque, restricted to be no greater than the account storage space available at Abiquiu Reservoir, minus the total potential release of San Juan water from El Vado Dam. The amount water in common storage accounts at El Vado Reservoir and Abiquiu Reservoir is computed with the user-defined TotalAccountStorageAvailable function in the General Compute Functions utility group. The total potential release of San Juan-Chama Project water from El Vado Dam is computed using the user-defined TotalPotentialDestinationReleaseElVado function in the General Compute Functions utility group. This function computes the maximum release as the sum of the following four parameters: the amount of San Juan water that could be released from El Vado Dam



for MRGCD determined using the user-defined MaxElVadoMRGCDSJRelease function in the ElVado Account Functions utility group, the total payback for debts to the river at Otowi or Albuquerque's account at Lake Abiquiu as computed using the user-defined CurrentTotalObjectDebtFlow function in the Exchange Functions utility group, the total amount for flow through accounts at El Vado Reservoir as determined with the user-defined TotalElVadoFlowThruAccounts function in the ElVado Account Functions utility group, and the total amount of water in storage accounts at El Vado Reservoir for accounts that also have storage accounts at Abiquiu Reservoir – not including MRGCD – minus the amount of that water to be used to payback debts to Otowi or Albuquerque restricted to be no greater than the account storage space at Abiquiu Reservoir as determined with the TotalAccountStorageAvailable function.

### **A.8.38. SetElVadoMRGCDDemand**

*Explanation:* This rule uses hypothetical simulation to determine the release from El Vado Dam required to meet the MRGCD demand at Abiquiu.

*Rule Execution:* If the MRGCD Demand time series slot in the ElVadoData data object for the current timestep is a NaN, the rule fires.

*Rule Logic:* The MRGCDDemand time series slot in the ElVadoData data object for the current timestep is set with the user-defined ComputeElVadoMRGCDDemand function in the ElVado Functions utility group. This function uses the predefined HypTargetSimWithStatus function to determine the required outflow from El Vado Dam (or the value for the Gage Inflow time series slot for the BlwElVado stream gage object) to meet a target inflow to Abiquiu Reservoir (or a value for the Outflow time series slot in the ElVadoToAbiquiuLocalInflow reach object) equal to the MRGCDDemand time series slot in the AbiquiuData data object for the next timestep (or the current timestep if that value is a NaN) as referenced with the user-defined GetAbiquiuMRGCDDemand function in the Get Data Functions utility group. The hypothetical simulation is completed for all the objects in the BelowElVado subbasin and uses the input minimum and maximum outflows from El Vado Dam of 0 and 5000 cfs, respectively, as bounds for the solution procedure. A tolerance of 2 cfs is allowed for the target flow.

### **A.8.39. Indian Storage Requirement Release First or Second Timestep**

*Explanation:* This rule computes and records the value for a call for P&P water in storage to meet the P&P demand at Cochiti Lake for the first and second timestep (The same logic is used for subsequent timesteps). Values are set in the future, so a separate rule is needed for the beginning of a simulation.

*Rule Execution:* If the current timestep is the Start Timestep or the next timestep after the Start Timestep and the value for the Indian Call time series slot in the Indian data object for the current timestep is a NaN, the rule fires.

*Rule Logic:* An exterior IF THEN ELSE statement is included in this rule to see if it is the Indian irrigation season as determined with the user-defined IndianIrrigationSeason function in the Date Functions utility group. The Indian irrigation season extends from March 1 through November 15. If it is NOT the Indian irrigation season, the value for the Indian Call time series slot in the Indian data object for the current timestep is set to 0.0 cfs. If it is the Indian irrigation season, the Indian call is set to the value in the IndianDemand time series slot in the MiddleValleyDemands data object for the current timestep minus the usable flow factor times the value for the Gage Outflow time series slot in the Embudo stream gage object for the current timestep all multiplied by  $(1 + (1 - 0.8))$  if that value is greater than zero. If that value is not greater than zero, the Indian call is set to zero. Refer to Figure A.24 for a flowchart that depicts the logic used for this rule.

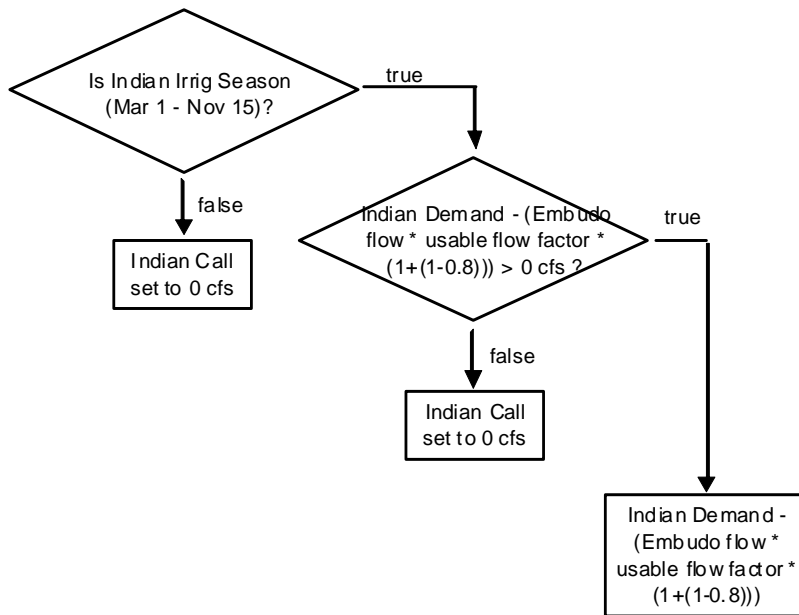


Figure A.24. Flow Chart Depicting the Calculation of Releases from P&P Storage

#### A.8.40. Indian Storage Requirement Release

*Explanation:* This rule sets the Indian call two days in advance (i.e. at the timestep equal to the current timestep plus two timesteps).

*Rule Execution:* If the value for the Indian Call time series slot in the Indian data object for the timestep equal to the current timestep plus two timesteps is a NaN, the rule fires.

*Rule Logic:* The logic for this rule (Figure A.24) is the exact same as the logic for the Indian Storage Requirement Release First or Second Timestep Rule. Refer to Figure A.25 for a screen capture of the rule policy language for this rule.

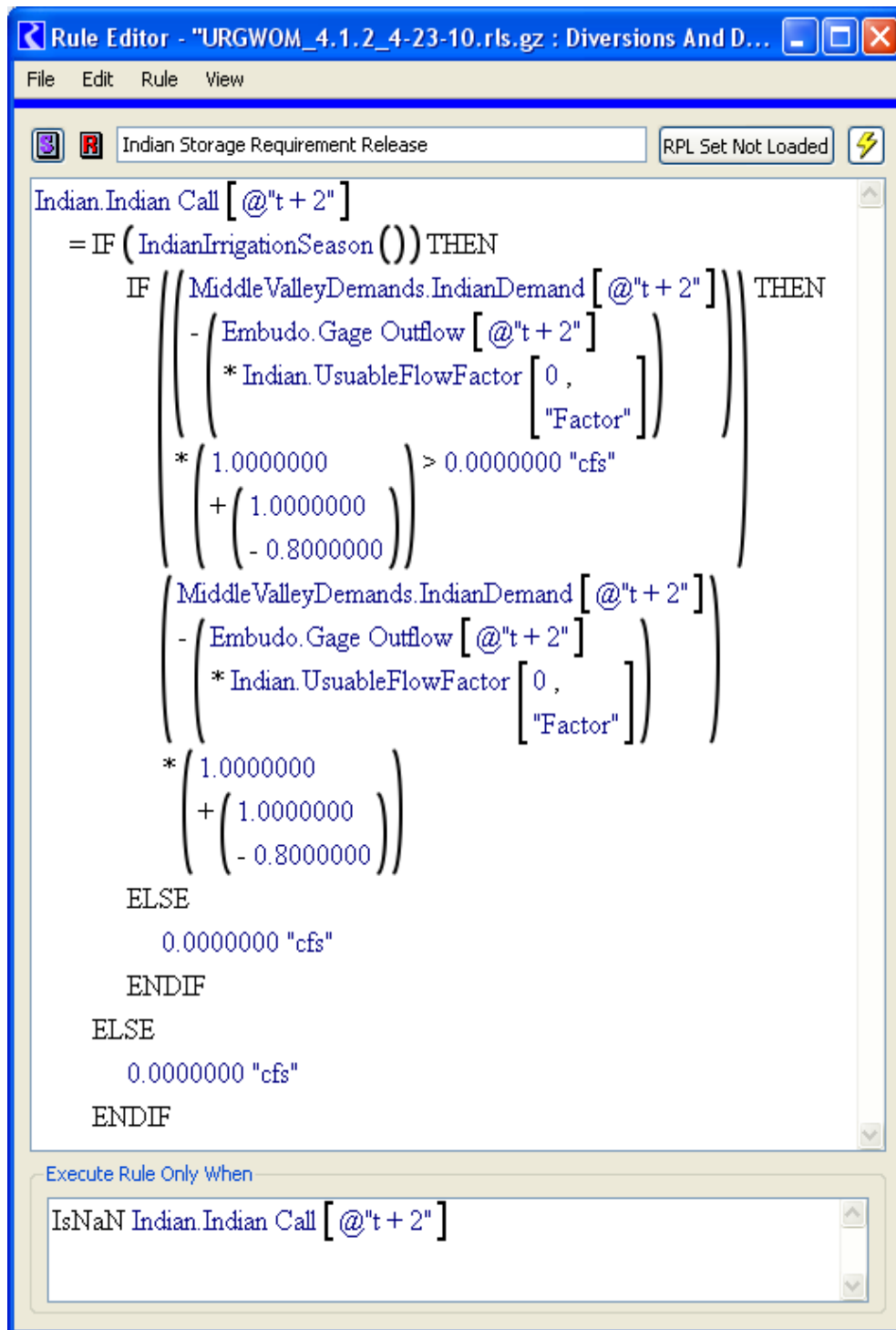


Figure A.25. Rule Policy Language for the Indian Storage Requirement Release Rule

#### **A.8.41. SetMinElVadoMRGCDRelease**

*Explanation:* This rule computes and records the value for the minimum release from El Vado Dam needed to meet the MRGCD demand after any contributions from letter water deliveries for other contractors are considered.

*Rule Execution:* If the value for the ComputedMinMRGCDRelease in the ElVadoData data object for the current timestep is a NaN, the rule fires.

*Rule Logic:* The value for the ComputedMinMRGCDRelease in the ElVadoData object for the current timestep is set using the user-defined ElVadoMRGCDRelease function in the ElVado Account Functions utility group. This function includes an IF THEN ELSE statement to see if the amount of water MRGCD has in its storage account at Abiquiu Reservoir is less than the maximum with consideration for a tolerance as entered to the MaxAccountStorageTolerance table slot in the AbiquiuData data object and referenced with the user-defined MaxAccountStorageTolerance function in the Get Data Functions utility group. If so, the release is set to the value for the MRGCDDemand time series slot in the ElVadoData data object for the current timestep minus the letter water adjustment for contractor letter water deliveries from El Vado Reservoir and minus the letter water adjustment for deliveries from Abiquiu Reservoir. If not, the release is set to the same value but reduces that result to correct for the extra water MRGCD has in its storage account at Abiquiu above the maximum.

The maximum amount of water MRGCD can have in its storage account at Abiquiu Reservoir is input to the MaxAccountStorage periodic slot in the AbiquiuData data object and is referenced with the user-defined MaxAccountStorage function in the Get Data Functions utility group. The letter water adjustments are computed by summing the payback debt for the OtowiPayback supplies while restricting the result to the maximum outflow of San Juan-Chama Project water from the reservoirs as determined with the user-defined MaximumSJOutflowHeron, MaximumSJOutflowElVado, and MaximumDeliveryRequestRelease functions in the Get Data Functions utility group. The letter water adjustment only considers the portion of letter water deliveries to payback MRGCD as estimated with the ScheduledDeliveryForMRGCDnotCompact function.

#### **A.8.42. ComputedMinAbiquiuRGRelease**

*Explanation:* This rule computes the minimum release of Rio Grande water from Abiquiu Dam as the input minimum release.

*Rule Execution:* If the value for the ComputedMinRGDemandRelease time series slot in the AbiquiuData data object for the current timestep is a NaN and the value for the MinRGDemandRelease time series slot in the AbiquiuData data object for the next timestep is a NaN, the rule fires.

*Rule Logic:* The ComputedMinRGDemandRelease time series slot in the AbiquiuData data object for the current timestep is set to the result from the user-defined MinRGOutflow function in the Get Data Functions utility group that references the value in the MinRGOutflow periodic slot in the AbiquiuData data object.

#### **A.8.43. SetMinAbiquiuRGRelease**

*Explanation:* This rule sets the minimum Rio Grande release for the current timestep and the next timestep equal to the computed minimum Rio Grande release for the current timestep.

*Rule Execution:* If the value for the MinRGDemandRelease time series slot in the AbiquiuData data object for the next timestep is a NaN, the rule fires.

*Rule Logic:* A FOREACH loop is used to assign the value for the MinRGDemandRelease time series slot in the AbiquiuData data object for the current timestep and the next timestep equal to the value of the ComputedMinRGDemandRelease time series slot in the AbiquiuData data object for the current timestep.

#### **A.8.44. AbiquiuTotalFlowToMeetTarget**

*Explanation:* This rule uses hypothetical simulation to determine the release from Abiquiu Dam required to meet the computed minimum flow needed at Cochiti Lake for all the Middle Valley targets (Refer to the SetCochitiMinimumFlow Rule). The calculations are not completed if there is no supplemental water available in storage or the estimated inflow to Cochiti Lake is high.

*Rule Execution:* If the value for the TotalFlowNeededToMeetTarget time series slot in the AbiquiuData data object for the current timestep is a NaN, the rule fires.

*Rule Logic:* An IF THEN ELSE statement is used to see if the sum of the storage in the Reclamation account at Heron, the Reclamation and Supplemental ESA accounts at El Vado, the Reclamation and Rio Grande Conservation accounts at Abiquiu and the Rio Grande Conservation account at Cochiti is greater than zero and if the EstimatedInflow to Cochiti is less than 2.5 times the value in the MRGCD time series slot in the MiddleValleyDemands data object for the current timestep or 2.0 times the value for the MinTargetForCentral time series slot in the Middle Valley Demands data object for the current timestep. If so, the TotalFlowNeededToMeetTarget time series slot in the AbiquiuData data object for the current timestep is set with the user-defined AbiquiuMinFlowsDemand function in the Demand Functions utility group. Otherwise the assignment is set to 0.0 cfs. The AbiquiuMinFlowsDemand function uses the predefined HypTargetSimWithStatus function to determine the required outflow from Abiquiu Dam (or the value for the Gage Inflow time series slot for the BlwAbiquiu stream gage object) to meet a target inflow to Cochiti Lake (or a value for the Outflow

time series slot in the OtowiToCochitiLocalInflow reach object) equal to the value for the MinimumFlow time series slot in the MiddleValleyDemands data object for the next timestep. The function completes the hypothetical simulation for all the objects in the BelowAbiquiu subbasin and uses the input minimum and maximum outflows from Abiquiu Dam of 0 and 1800 cfs, respectively (the channel capacity below Abiquiu Dam is 1800 cfs) as bounds for the solution. A tolerance of 5 cfs is allowed for the target flow.

## **A.9. Preparatory Heron And ElVado**

The rules in this policy group set preliminary values for different parameters that are referenced later when ultimately determining the release from Heron Dam and El Vado Dam.

### **A.9.1. SetRelinquishedNMCredits**

*Explanation:* If the policy for relinquished Compact credits is being modeled, this rule computes a magnitude for the relinquished credits on the input relinquishment date. If a value for a relinquishment is not input then the relinquished credit is computed. If the input threshold Compact credit before a relinquishment will occur is exceeded, the relinquishment is computed as the difference in the Compact credit and an input target Credit after a relinquishment.

*Rule Execution:* The rule fires if the value input to the TriggerModelRelinquishedCredits scalar slot in the ElVadoData data object is greater than 0.1 as checked with the user-defined ModelRelinquishedCredits function and the current timestep is equal to the date for a relinquishment as input in the RelinquishedCreditsTriggers table slot and checked with the user-defined RelinquishmentDate function and the RelinquishedNMCredits time series slot in the ElVadoData data object is a NaN for the current timestep.

*Rule Logic:* If the storage in the NMCredit account at Elephant Butte Reservoir at the end of the previous year (or the initial timestep if sooner) as identified with the user-defined CreditWater function in the RelinquishedNMCreditFunctions utility group is less than the value in the ThresholdForRelinquishment column in the RelinquishedCreditsTriggers table slot in the ElVadoData data object, the value for the RelinquishedNMCredits time series slot for the current timestep is set to the difference between and the storage in the NMCredit account at Elephant Butte Reservoir and the value in the CreditAfterRelinquishment column in the RelinquishedCreditsTriggers table slot in the ElVadoData data object.

### **A.9.2. UpdateEmergencyDroughtStorageAllocations**

*Explanation:* This rule updates the tracked allocations for storage of Emergency Drought water at El Vado Reservoir for MRGCD, ESA purposes, and municipalities each timestep

during a simulation. Allocations at the Start Timestep are set based on input start values. Allocations include any water already in storage. Allocations are updated each day for any used allocation reflected by water released from the corresponding storage account and increased for any new relinquished credits with the total allocation split between the three uses based on input percentages.

*Rule Execution:* If the values in the MRGCDDroughtAllocation, SupplementalESAAllocation, and MunicipalitiesAllocation time series slots in the ElVadoData data object are all NaNs for the current timestep, the rule fires.

*Rule Logic:* If it is the Start Timestep, the value for the MRGCDDroughtAllocation time series slot is set to the initial input allocation in the MRGCDDrought row in the RelinquishedNMCreditAllocations table slot in the ElVadoData data object. Otherwise, the allocation is calculated as the value in the MRGCDDroughtAllocation series slot in the ElVadoData data object for the previous timestep minus the volume of the release from the account at the previous timestep plus any account gain loss plus any new allocation resulting from relinquished credits checked as follows. If the value input to the TriggerModelRelinquishedCredits scalar slot in the ElVadoData data object is greater than 0.1 as checked with the user-defined ModelRelinquishedCredits function and the current timestep is equal to the date for a relinquishment as input in the RelinquishedCreditsTriggers table slot and checked with the user-defined RelinquishmentDate function and the value in the RelinquishedNMCredits series slot in the ElVadoData data object for the current timestep is greater than zero, the relinquished credit for the current timestep multiplied by the proportion to allocate to MRGCD as input to the Proportion column and MRGCDDrought row in the RelinquishedNMCreditAllocations table slot in the ElVadoData data object is added to the allocation. The allocations for ESA purposes are computed in the same manner with reference to the inputs for SupplementalESA, and allocations for municipalities are also computed in the same manner; although, storage for municipalities is not actually modeled, so no releases from a municipalities account are considered.

### **A.9.3. SetCumulativeCochitRecPoolRelease**

*Explanation:* This rule tallies the amount of water released from Heron Reservoir for the Cochiti recreation pool. This tally is reset based on input start dates. The tally is kept to assure the transfers of water for the Cochiti Rec Pool do not exceed the input limits for designated periods within the calendar year. The cumulative delivery volumes input to the model allow for the deliveries to be controlled by the model user.

*Rule Execution:* If it is after the Start Timestep and the value in the CumulativeCochitiRecPoolRelease time series slot in the HeronData data object for the previous timestep is a NaN, the rule fires.

*Rule Logic:* A WITH DO loop is used with the user-defined PeriodicSlotStartDateList function in the Account Lists utility group to create a list of start dates in the

CochitiRecPoolMaxVolume periodic slot in the HeronData data object. The predefined DatesInPeriod function is used in the PeriodicSlotStartDateList function. An IF THEN ELSE statement is used in the rule to determine if the previous date is one of the start dates in the CochitiRecPoolMaxVolume periodic slot. If so, the CumulativeCochitiRecPoolRelease time series slot in the HeronData data object for the previous timestep is set to 0.0 acre-ft. If not, this slot is set to the value for the CumulativeCochitiRecPoolRelease time series slot for the timestep before the previous timestep plus the volumetric sum of all the supplies for releases out of Heron Reservoir with a release type of CochitiRecPool. The predefined FlowToVolume function is used to convert the sum of supplies to a volume. The user-defined SumSuppliesAtDate function in the Account Summing Functions utility group is used to sum the list of supplies generated with the user-defined AllSuppliesByReleaseTypeDestination function in the Account Lists utility group. Refer to Figure A.26 for a flowchart that depicts the logic used for this rule.

#### **A.9.4. SetCumulativeAccountFillRelease**

*Explanation:* This rule tallies the amount of water released from Heron Reservoir to fill allocated downstream storage space for contractors for San Juan-Chama Project water. The cumulative account fill volumes input to the model for designated periods allow for the model user to control the movement of contractor water to downstream allocated storage space. The tally is reset at the input start dates for separate designated periods.

*Rule Execution:* If it is after the Start Timestep and the CumulativeAccountFillRelease time series slot in the HeronData data object for the previous timestep is a NaN, the rule fires. The latter criterion ensures that the rule only fires once per timestep.

*Rule Logic:* A WITH DO loop is used with the user-defined PeriodicSlotStartDateList function in the Account Lists utility group to create a list of start dates in the AccountFillMaxVolume periodic slot in the HeronData data object. The predefined DatesInPeriod function is used in the PeriodicSlotStartDateList function. An IF THEN ELSE statement is used in the rule to determine IF the previous date is one of the start dates in the AccountFillMaxVolume periodic slot. If so, the CumulativeAccountFillRelease time series slot in the HeronData data object for the previous timestep is set to 0.0 acre-ft. If not, this slot is set to the value for the CumulativeAccountFillRelease time series slot for the timestep before the previous timestep plus the volumetric sum of all the supplies at Heron Reservoir with a release type of AccountFill. The predefined FlowToVolume function is used to convert the sum of supplies to a volume. The user-defined SumSuppliesAtDate function in the Account Summing Functions utility group is used to sum the list of supplies generated with the user-defined AllSuppliesByReleaseTypeDestination function in the Account Lists utility group. Refer to Figure A.26 for a flowchart that shows the logic used for this rule.



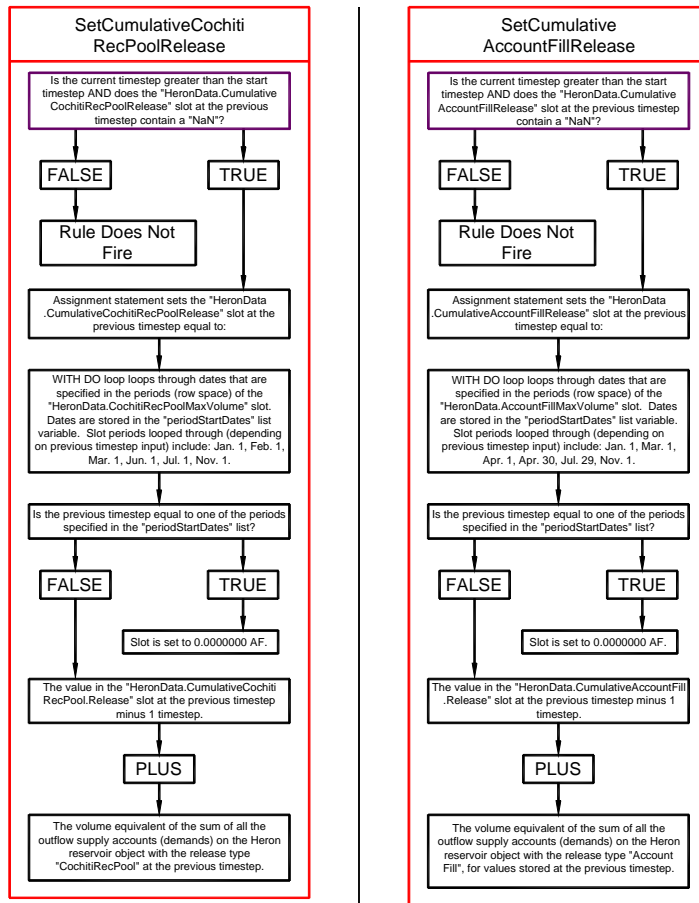


Figure A.26. Flowcharts for the SetCumulativeCochitiRecPoolRelease and SetCumulativeAccountFillRelease Rules

### A.9.5. SetSanJuanContractorAllocationsJuly1

*Explanation:* This rule is used to allocate water in the federal pool at Heron Reservoir to contractors for San Juan-Chama Project water on July 1. Additional allocations for a year would be made on July 1<sup>st</sup> (after the runoff and the additional inflows from the Azotea tunnel to the federal pool) if the supply was not available to make full allocations on January 1<sup>st</sup> with the SetSanJuanContractorAllocations rule.

*Rule Execution:* If the current timestep is July 1, the rule fires.

*Rule Logic:* A FOREACH loop is used to set the supplies to make allocations of San Juan water in Heron Reservoir to contractors on July 1. The list of supplies is created using the user-defined AccountsByWaterTypeWaterOwners function in the Account Lists utility group. The supplies are set using the user-defined Compute1JulSanJuanContractorsAllocations function in the Heron Account Functions utility group.

Within the function, an exterior IF THEN ELSE statement is used to see if water is available to allocate. The sum of the Begin Year Allocations for all the accounts as determined with the `1JanSanJuanAllocations` function in the Heron Account Functions utility group is checked to see if it is less than the total contract allocation for all the contractors in the `SanJuanContractorAllocations` table slot in the `HeronData` data object, where rows 0 to 15 in the table are checked for the Water Operations Model and rows 0 to 3 for the Planning Model and where the model being used is identified with the user-defined `IsWaterOpsRunNotPlanningRun` function. Also, the storage in the federal pool at the previous timestep as determined with the user-defined `PreviousAccountStorage` function in the General Account Functions utility group is checked to assure it is greater than 1000 acre-ft. If these checks are satisfied, allocations are made. Otherwise, the supplies are set to 0.0 cfs. An interior IF THEN ELSE statement is used with the same functions to see if the amount of water in the federal pool at the previous timestep plus the sum of the Jan 1 allocations to contractors is greater than the total contract allocation. If so, the supplies are set to the input allocations in the `SanJuanContractorAllocations` table slot on the `HeronData` data object minus the Begin Year Allocation for the corresponding account. If not, the supplies are set to the same value but multiplied by the ratio of the contract allocation for the account divided by the sum of the total contract allocations. The predefined `VolumeToFlow` function is used to convert the allocation volumes to a flow.

#### **A.9.6. SetSanJuanContractorAllocations**

*Explanation:* This rule is used to allocate water in the federal pool at Heron Reservoir to contractors for San Juan-Chama Project water on January 1 of each year. Allocations are made proportionally up to the full annual allocation amounts for each contractor. Contractor water from the previous year allocation still in storage at Heron Reservoir is reverted back to the Federal pool but only for contractors that do not have downstream allocated storage space.

*Rule Execution:* If the current timestep is January 1 and the Begin Year Allocation for the Albuquerque account at Heron Reservoir is a NaN, the rule fires.

*Rule Logic:* A FOREACH loop is used to set the Begin Year Allocation for the storage accounts at Heron Reservoir for each contractor for San Juan-Chama Project water on January 1. The list of accounts is created using the user-defined `AccountsByWaterTypeWaterOwners` function in the Account Lists utility group. The Begin Year Allocation values are set using the `Compute1JanSanJuanContractorAllocations` function in the Heron Account Functions utility group. Refer to Figure A.27 for a screen capture of the RiverWare rule policy language for this function.

Within the function, an exterior IF THEN ELSE statement is used in this function to see if water is available to allocate. The Begin Year Allocation for the corresponding

account at Heron Reservoir is checked to see that it is a NaN and the storage in the federal pool at the previous timestep as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group is checked to see if it is greater than or equal to the total contract allocation for all the contractors in the SanJuanContractorAllocations table slot in the HeronData data object, where rows 0 to 15 in the table are checked for the Water Operations Model and rows 0 to 3 for the Planning Model and where the model being used is identified with the user-defined IsWaterOpsRunNotPlanningRun function. If these checks are satisfied, allocations are made. Otherwise, the supplies are set to 0.0 cfs. Allocations are set to the amount of water in the federal pool at the previous timestep multiplied by the input contract allocation for the corresponding account divided by the sum of all contract allocations.

An additional assignment statement is included in this rule to set the Begin Year Allocation for the FederalSanJuan account at Lake Heron to the sum of the storage for all the accounts other than the accounts that have storage for San Juan water at Heron, El Vado, or Abiquiu Reservoir. The user-defined StringListsNonIntersect and StringListsUnion functions in the General List Functions utility group are used to create the list of storage accounts to sum, and the SumAccountStorage function in the Account Summing Functions utility group is used to complete the summation.

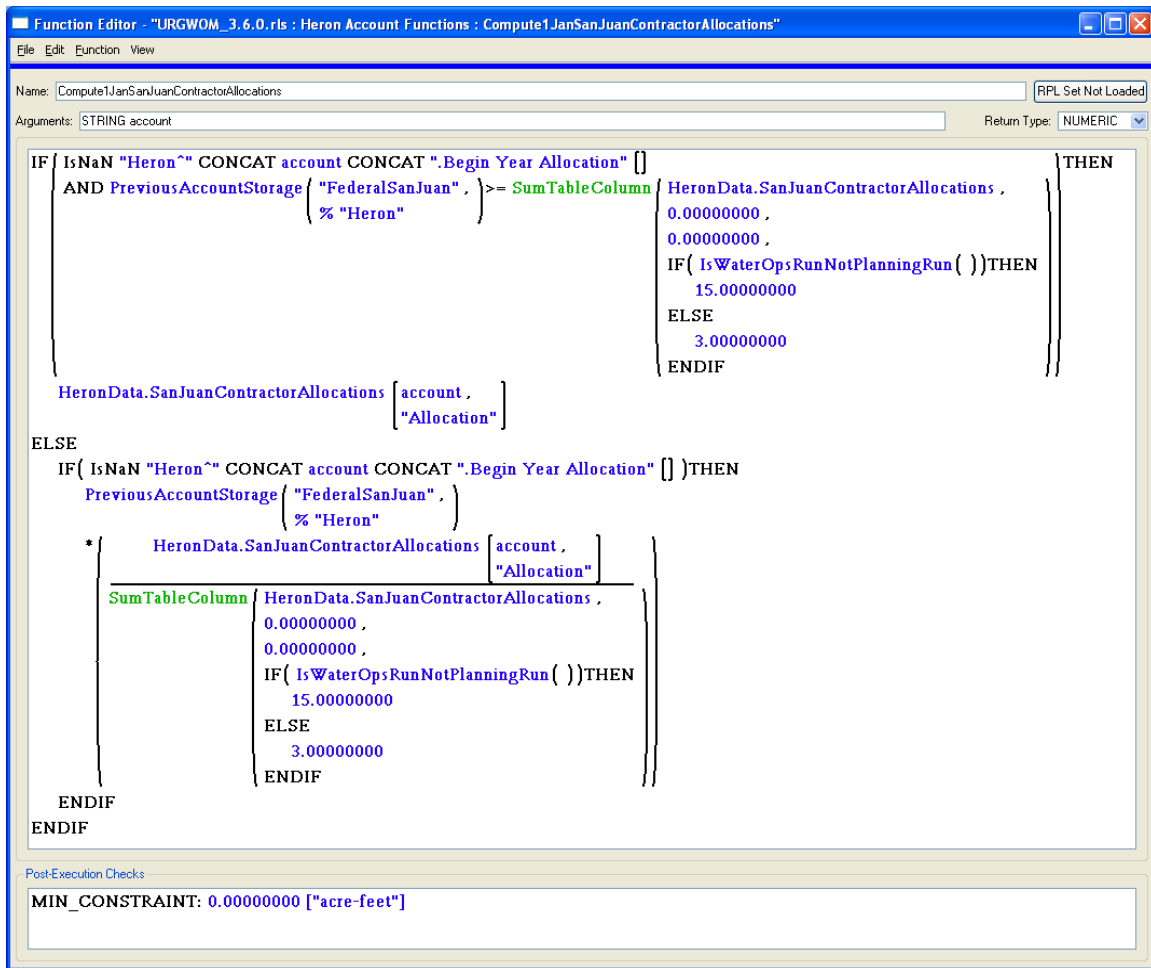


Figure A.27. Rule Policy Language for the Compute1JanSanJuanContractorAllocations Function used in the SetSanJuanContractorAllocations Rule

### A.9.7. SetPercentRGRelease

*Explanation:* When Article VII is not in effect, El Vado Reservoir will be filled during the runoff, and the approach for doing this entails storing all inflows (not needed to meet downstream demands) up to a specific elevation and then bypassing a computed percent of the inflows thereafter to exactly fill the reservoir by an input target fill date. This rule computes the percentage of the Rio Grande inflow that will be released. The percent release is set to zero (i.e. all inflows not needed to meet downstream demands are stored) until the storage reaches an input percentage of the maximum storage. Essentially, all inflows are stored up to a specific elevation, and a percentage of inflows are bypassed thereafter to fill the reservoir on the input fill date.

*Rule Execution:* This rule fires if the following conditions are satisfied. The value for the PercentRGRelease time series slot in the ElVadoData data object for the current timestep is a NaN and the current timestep is before the target fill date as determined with the

user-defined ElVadoTargetFillDate function in the ElVado Functions utility group (Refer to Section A.9.7.1 for more information on the TargetFillDate function.)

Also, the Storage for the El Vado storage reservoir object at the previous timestep as determined with the user-defined PreviousStorage function is checked to assure it is greater than or equal to a input percentage of the storage corresponding with the result from the user-defined ElVadoTargetFillElevation function in the Get Data Functions utility group, and the previous storage at El Vado Reservoir plus the forecasted inflow as determined with the user-defined ForecastElVadoInflow function in the Estimated Inflow Functions utility group is checked to assure it is greater than or equal to 90% of the storage corresponding to the result from the ElVadoTargetFillElevation function. Also, the stipulations for Compact VII of the Compact cannot have been in effect since the beginning of the year as determined with the user-defined GetMaxCompactVIISwitchJan1ThruCurrentTimestep function in the ElVado Compact Functions utility group. The value in the TargetElevation periodic slot corresponding to the current timestep as identified with the user-defined TargetElevation function in the Get Data Functions utility group is checked to assure it is greater than the minimum target elevation from the same periodic slot as identified with the user-defined MinTargetElevation function in the General Compute Functions utility group.

*Rule Logic:* A FOREACH loop is used to set the value for the PercentRGRelease time series slot in the ElVadoData data object for the current timestep plus the next six timesteps. The value is set to the result from the user-defined ComputePercentElVadoRGRelease function in the General Compute Functions utility group. This function sets the percent release to 1.0 minus the following quantity: the target storage minus the account storage adjustment minus the sum of storage accounts at El Vado for Rio Grande water all divided by the forecasted inflow to El Vado Reservoir. The target storage is referenced using the user-defined TargetStorage function in the Get Data Functions utility group which uses the Target Elevation function to reference the TargetElevation periodic slot in the ElVadoData data object. That value is converted to a storage with the predefined ElevationToStorage function. Refer to the Section A.9.7.2 for more discussion of the function used to compute the account storage adjustment. The user-defined SumAccountStorageAtGivenDate function in the Account Summing Functions utility group is used to sum the accounts at El Vado Reservoir for Rio Grande water as identified using the predefined AccountNamesByWaterType function. The forecasted inflow to El Vado Reservoir for the current timestep to the timestep for the target elevation is determined with the user-defined ForecastElVadoInflow function in the Estimated Inflow Functions utility group. This function utilizes the user-defined EstimateElVadoInflow function in the Estimated Inflow Functions utility group which references the input values in the Inflow2 time series slot in the AzoteaWillow confluence object and the Local Inflow time series slot in the ElVadoLocalInflow reach object. A percent forecast error is also applied if it is not a NaN. Refer to Figures A.28 and A.29 for a flowchart that depicts the logic used in this rule.

#### A.9.7.1 TargetFillDate

*Function Logic:* The TargetFillDate function references the ComputeTargetFillDate function in the General Compute Functions utility group to compute the target fill date for El Vado Reservoir which is essentially the date that corresponds with the highest input reservoir target elevation. The ComputeTargetFillDate function includes an exterior WITH DO loop that uses the user-defined PeriodicSlotDateList function in the Account Lists utility group to create a list of dates in the TargetElevation periodic slot in the ElVadoData data object. A second WITH DO loop is included that uses the user-defined TargetElevation function in the Get Data Function utility group to create a list of elevations in the TargetElevation periodic slot. A third WITH DO loop identifies the maximum elevation from that list using the predefined MaxItem function. A fourth WITH DO loop uses the user-defined NumericListElement function in the General List Functions utility group with the GET DATETIME @INDEX command to identify the date that corresponds with the max elevation.

#### A.9.7.2. ElVadoAccountStorageAdjustment

*Explanation:* The user-defined ElVadoAccountStorageAdjustment function in the ElVado Functions utility group is used to compute the account storage adjustment. Four separate equations are included to compute the adjustment depending on specified criteria.

*Function Logic:* 1. If the current timestep is before the target fill date and the ElVadoIsPriority function is true (Refer to Section A.9.7.3), the storage adjustment is set to the sum of the San Juan water in storage accounts at El Vado Reservoir as determined with the predefined SumAccountSlotsByWaterType function plus the volumetric sum of the supplies with an MRGCDPaybacks type and an MRGCD\_ElVado destination identified with the user-defined AllSuppliesByReleaseTypeDestination function and summed using the user-defined SumSuppliesAtDate function plus the maximum of zero or the difference in the value for the AccountFillMaxVolume time periodic slot in the HeronData data object referenced using the user-defined AccountFillMaxVolume function in the Get Data Functions utility group and the value for the CumulativeAccountFillRelease time series slot in the HeronData data object for the previous timestep.

2. If the current timestep is before the target fill date and the ElVadoIsPriority function is false, the storage adjustment is the sum of the San Juan and Reclamation storage accounts for San Juan water at El Vado Reservoir identified with the user-defined AccountsByWaterTypeWaterOwners function and summed using the user-defined SumAccountStorage function plus the volumetric sum of the supplies with an MRGCDPaybacks type and an MRGCD\_ElVado destination identified with the user-defined AllSuppliesByReleaseTypeDestination function and summed using the user-defined SumSuppliesAtDate function.

3. If the current timestep is after the target fill date and the CurrentYearIsWaiverYear function is true (Refer to Section A.9.7.4), the storage adjustment is the sum of the San Juan and Reclamation storage accounts for San Juan water at El Vado Reservoir

identified with the user-defined AccountsByWaterTypeWaterOwners function and summed using the user-defined SumAccountStorage function plus the sum of the values in the AlbuquerqueFromMRGCDLoan time series slot in the HeronData data object for the current timestep to the date corresponding to the target elevation summed using the predefined SumSlot function.

4. If the current timestep is after the target fill date and the CurrentYearIsWaiverYear function is false, the storage adjustment is the sum of the San Juan and Reclamation storage accounts for San Juan water at El Vado Reservoir identified with the user-defined AccountsByWaterTypeWaterOwners function and summed using the user-defined SumAccountStorage function plus the sum of the values in the AlbuquerqueFromMRGCDLoan time series slot in the HeronData data object for the current timestep to the date corresponding to the target elevation summed using the predefined SumSlot Function plus the sum of storage accounts at Heron Reservoir that do not have downstream storage accounts and minus the sum of the storage accounts for the CochitiRecPool.

#### A.9.7.3. ElVadoIsPriority

*Function Logic::* The ElVadoIsPriority function in the General Boolean Functions utility group uses the PrioritizedReservoirs function in the Priority Functions utility group to create a list of the reservoirs in the ReservoirPriority periodic slot in the HeronData data object which are then sorted based on the input priority using the predefined SortPairsAscending function (The values in the ReservoirPriority periodic slot are referenced using the user-defined ReservoirPriority function in the Get Data Functions utility group).

#### A.9.7.4. CurrentYearIsWaiverYear

*Function Logic:* The user-defined CurrentYearIsWaiverYear function in the General Boolean Functions utility group is used to check to see if the input WaiverSwitch time series slot in the HeronData data object for December 31 of the current year is a 1. The WaiverSwitch time series slot is referenced using the user-defined CurrentYearWaiverSwitch function in the Get Data Functions utility group.

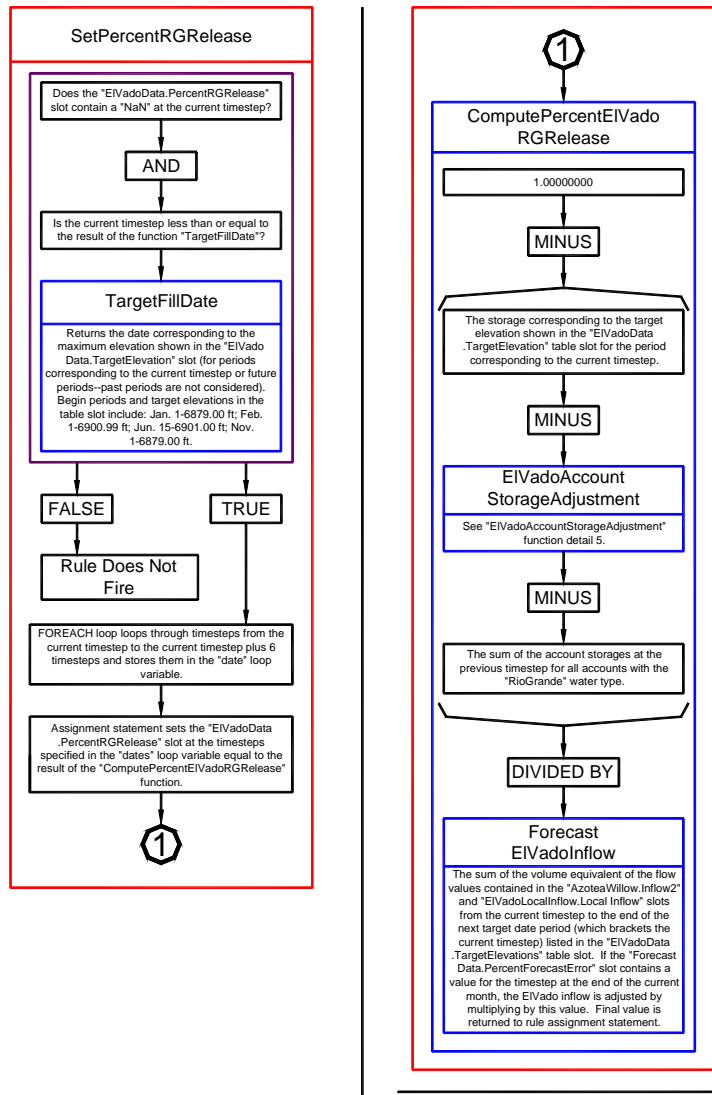


Figure A.28. Flowchart for the SetPercentRGRelease Rule



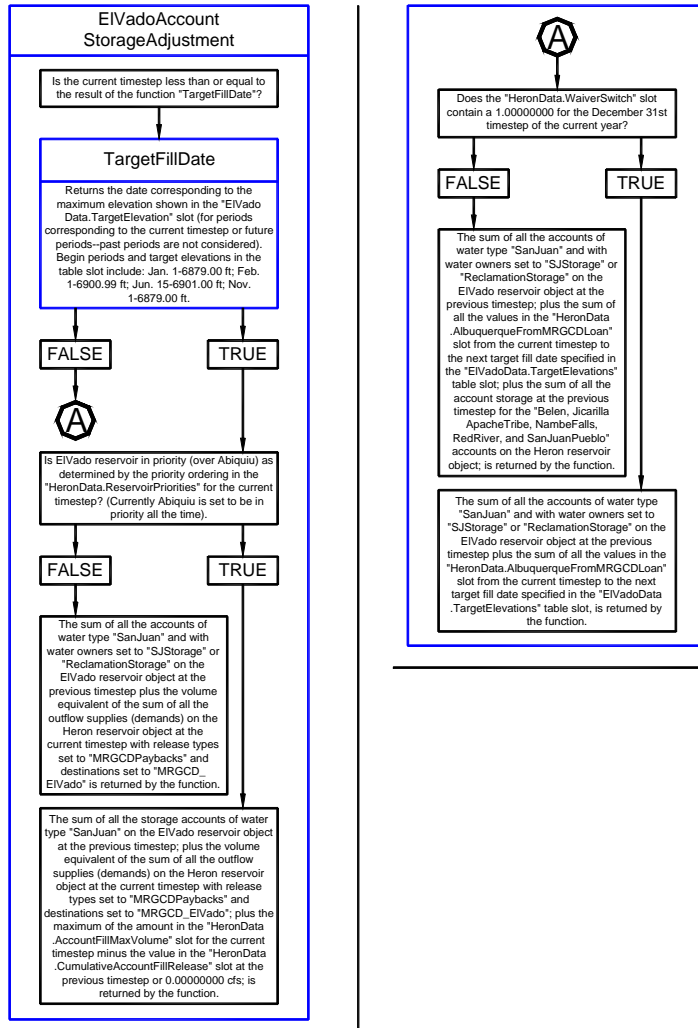


Figure A.29. Flowchart for the SetPercentRGRelease Rule (continued)

### A.9.8. SetElVadoIndianStorageReqAprilAndMarch

**Explanation:** This rule sets the Indian storage requirement at El Vado Reservoir on March 1 or April 1 or the Start Timestep if it is in March or April. The storage requirement is calculated based on the amount of water that would be needed to meet the Indian Demand in the Middle Valley during the irrigation season with consideration for natural flows from the mainstem.

**Rule Execution:** If the IndianStorageReq time series slot in the ElVadoData data object for the last day of the current month is a NaN and the current timestep is either April 1 or March 1 or the start timestep and between March 1 and April 30 (inclusive), the rule fires.

*Rule Logic:* The IndianStorageReq time series slot in the ElVadoData data object for the last timestep in the current month is set as the sum of the Indian storage requirement plus the dead storage at El Vado Reservoir. The dead storage is determined using the user-defined DeadStorage function in the Get Data Functions utility group based on the highest reservoir stage with zero outflow in the Max Release table for the ElVado storage reservoir object which is converted to a storage using the predefined ElevationToStorage function.

The user-defined ComputeIndianStorageReq and ComputeMonthlyIndianStorageReq functions in the Indian Storage Functions utility group are used to create a list of the monthly Indian storage requirements that are summed with the user-defined SumList function. The ComputeMonthlyIndianStorageReq function computes the requirement as the maximum of zero or the value in the IndianDemand table slot for the corresponding month as referenced with the user-defined IndianDemand function in the Get Data Functions utility group minus the result from the ElVadoUsableFlow function in the Indian Storage Functions utility group all divided by the value for the IndianStorageAdjFactor table slot in the Indian data object and multiplied by the quantity of 1.0 plus the input value in the FuturePossibilitiesIndianStorageRequirementAdjustment table slot in the Indian data object. The ElVadoUsableFlow function multiplies the result from the user-defined ComputeSupplyAtOtwi function by the corresponding input value in the UsableFlowFactor [sic] table slot in the Indian data object referenced using the user-defined LookupUsableFlowFactor [sic] function in the Indian Storage Functions utility group. Refer to Figures A.30 for a diagram that outlines the steps for computing the Indian Storage Requirement.

#### A.9.8.1. ComputeSupplyAtOtwi

*Function Logic:* The ComputeSupplyAtOtwi function computes the Otowi supply as a function of the month. The value is set to the input value in the LowRecordLateSummer table slot in the Indian data object for corresponding month plus the result from the ElVadoRunoffLeft function multiplied by the value in the MonthPercent table slot in the ElVadoData data object for the current month referenced with the user-defined GetMonthPercent function in the Indian Storage Functions utility group all minus the value in the AnticipStorage table slot in the Indian data object for the corresponding month.

If the month is March, the result from the ElVadoRunoffLeft function is set to the value in the OtowiForecast time series slot in the Indian data object for the current year (Refer to the ComputeOtowiForecast Rule) referenced using the InputElVadoForecast function in the Indian Storage Functions utility group multiplied by the value in the ForecastFactors table slot in the ElVadoData data object for the corresponding month referenced using the user-defined ElVadoForecastFactor function in the Indian Storage Functions utility group. If the month is a month between April and July (inclusive), the result from the user-defined RealizedOtowiForecast function in the Indian Storage

Functions utility group is subtracted from the result from the InputElVadoForecast function before multiplying by the result from the ElVadoForecastFactor.

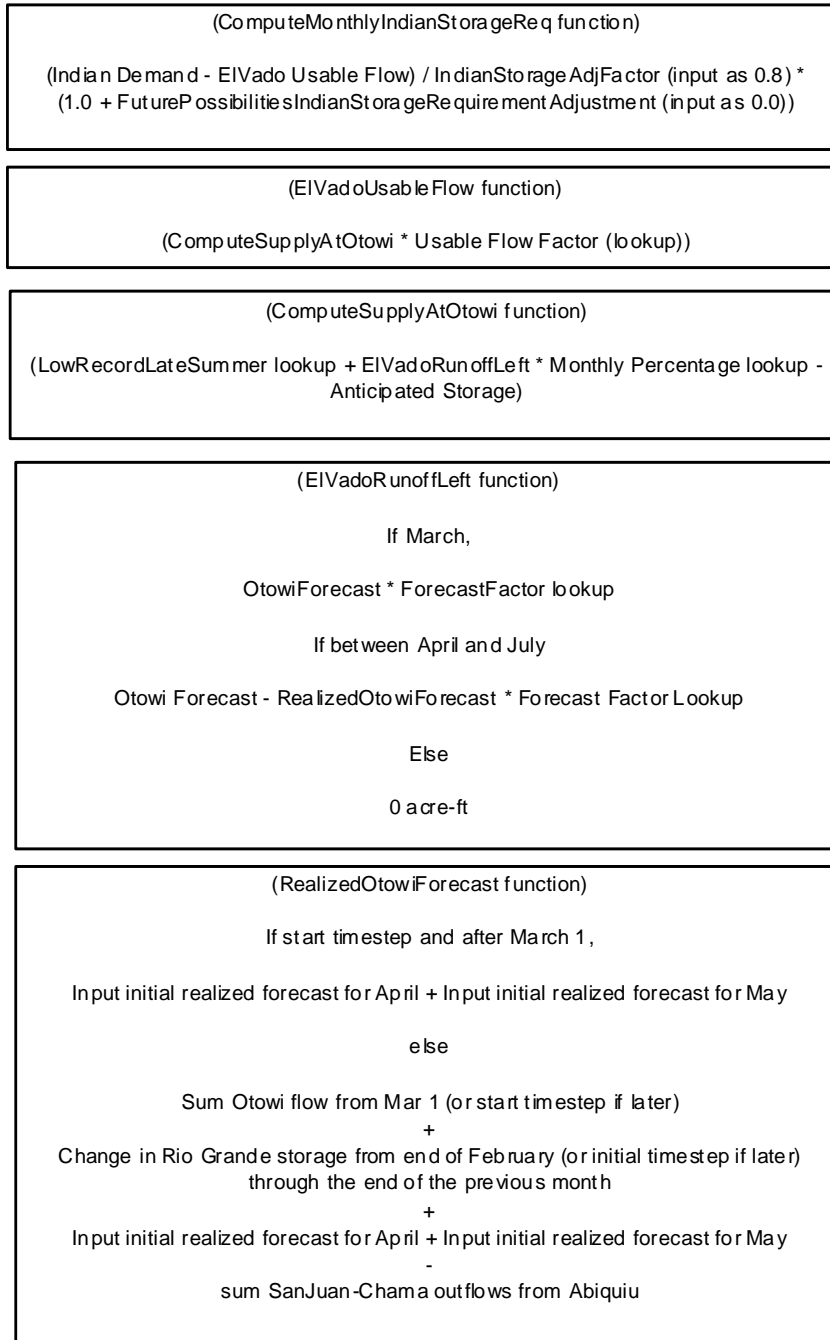


Figure A.30. Steps for Computing the Indian Storage Requirement

### A.9.9. SetElVadoIndianStorageReqAfterApril

*Explanation:* This rule sets the Indian storage requirement on the 1<sup>st</sup> of each month starting in May, or the start timestep if it is after May 1, to October (inclusive) using the same method used in the SetElVadoIndianStorageReqAprilAndMarch Rule for March and April.

*Rule Execution:* If the IndianStorageReq time series slot in the ElVadoData data object for the last day of the current month is a NaN and the current timestep is either May 1 or the start timestep and after April, the rule fires.

*Rule Logic:* The calculations in this rule for the Indian storage requirement use the same functions used in the SetElVadoIndianStorageReqAprilAndMarch Rule, but a FOREACH loop is used to set the values for each month from May to October. A WITH DO loop is used with a FOR WITH DO loop to sum the results for the storage requirement for each month up to October. A second WITH DO loop is used with a FOR WITH DO loop to add the dead storage for each month. Refer to the discussion for the SetElVadoIndianStorageReqAprilAndMarch Rule for more details on the functions and equations used to compute the Indian storage requirement.

#### **A.9.10. ZeroElVadoIndianStorageReq**

*Explanation:* This rule sets the Indian storage requirement to zero if it is a NaN. So, if for some reason a value was not set by the SetElVadoIndianStorageReqAprilAndMarch or SetElVadoIndianStorageReqAfterApril Rules, the Indian storage requirement is set to zero.

*Rule Execution:* If the IndianStorageReq time series slot in the ElVadoData data object for the last day of the current month is a NaN, the rule fires.

*Rule Logic:* An assignment statement is used to set the value for the IndianStorageReq time series slot in the ElVadoData data object for the last day of the current month to 0.0 acre-ft.

#### **A.9.11. ComputeHeronRGRelease**

*Explanation:* This rule computes the release of Rio Grande water from Heron Dam. Rio Grande water is effectively bypassed at Heron Dam, but it would not be practical to adjust the gates daily to bypass inflows, so actual operations entail setting releases for periodic evacuations of Rio Grande water in storage with consideration for inflows. The calculation in the rules is different depending on whether it is the first half of the month or the second half of the month. The release of Rio Grande water is set on Mondays during the first half of the month and set on Mondays and Thursdays during the second half of the month. Set releases are maintained until the next date to reset the release. The computed release includes consideration for the unregulated spill and the input maximum

storage of Rio Grande water. A Rio Grande base inflow and seepage from Heron Reservoir are considered at the first half of the month.

*Rule Execution:* If the value for the RGOutflow time series slot in the HeronData data object for the current timestep is a NaN, the rule fires.

*Rule Logic:* The value for the RGOutflow time series slot in the HeronData data object for the current timestep is set using the user-defined HeronRGOutflow function in the Heron Functions utility group. This function includes an IF THEN ELSE statement so the release can be computed with two separate equations depending on whether it is the first half of the current month (the 15<sup>th</sup> of the month or earlier) or the second half of the month (after the 15<sup>th</sup>) as determined with the user-defined IsFirstHalfOfMonth and IsSecondHalfOfMonth functions in the Date Functions utility group. Also, for both cases, the user-defined IsHeronRGRelGreaterThanZero function in the Heron Functions utility group is checked. The result from this function is TRUE if the value for the SJOutflow time series slot in the HeronData data object for the current timestep is greater than zero or the storage in the Rio Grande storage account at Heron Reservoir for the previous timestep is greater than the input value in the HeronMinRGStorageForRel table slot in the HeronData data object or the storage in the Rio Grande storage account at Heron Reservoir for the previous timestep is greater than the input value in the MaxRGStorage table slot in the HeronData data object or the unregulated spill as determined with the predefined ElevationToUnregulatedSpill function is greater than zero AND the value for the Surface Ice Coverage time series slot in the Heron storage reservoir object for the current timestep is not greater than the input value for the IceCoverageThreshold table slot in the HeronData data object as determined with the user-defined HeronHasFullIceCover function in the Heron Functions utility group.

For the first half of the month, the release is computed with the user-defined HeronFirstOfMonthRGRelease function in the Heron Functions utility group. If the unregulated spill as determined with predefined ElevationToUnregulatedSpill function is greater than zero then if the current timestep is a Monday, the release is set to the unregulated spill plus the average daily flow for the input value for the RGBaseFlow table slot for the current month referenced with the user-defined HeronRGBaseFlow function in the Heron Functions utility group plus the storage in the Rio Grande storage account at Heron Reservoir for the previous timestep divided by seven days. Otherwise if the value for the Storage time series slot in the Heron storage reservoir object for the previous day is greater than the storage corresponding to the input MaxElevation in the ReservoirData table slot in the HeronData data object referenced using the user-defined MaxStorage function in the Get Data Functions utility group and the current timestep is not a Monday, the release is set to the unregulated spill plus the additional release required to reduce the storage to the maximum storage or the maximum outflow if it is lower. Otherwise, the release is set to the unregulated spill plus the Outflow for the Rio Grande storage account at Lake Heron for the previous timestep minus the value for the Seepage time series slot in the Heron storage reservoir object for the previous timestep. If the unregulated spill is not greater than zero, the release is set using all the same equations under the same criteria except the unregulated spill is not included.

For the second half of the month, the release is computed in the same manner except if there is an unregulated spill, the additional release of water in the Rio Grande storage account is split between a Monday and Thursday release. Also, the release is not checked to assure that the input RGBaseFlow is maintained. Also, on other days, the value for the seepage from Heron Reservoir is not released with the Outflow for the Rio Grande storage account. If the unregulated spill is not greater than zero, the releases are set with the same equations except the unregulated spill is not included.

The HeronRGOutflow function includes a post execution constraint to assure the result does not exceed the maximum of either the input value in the MaxRGOutflow periodic slot or the unregulated spill. Refer to Figures A.31 through A.34 for a flowchart that depicts the logic and calculations for this rule.

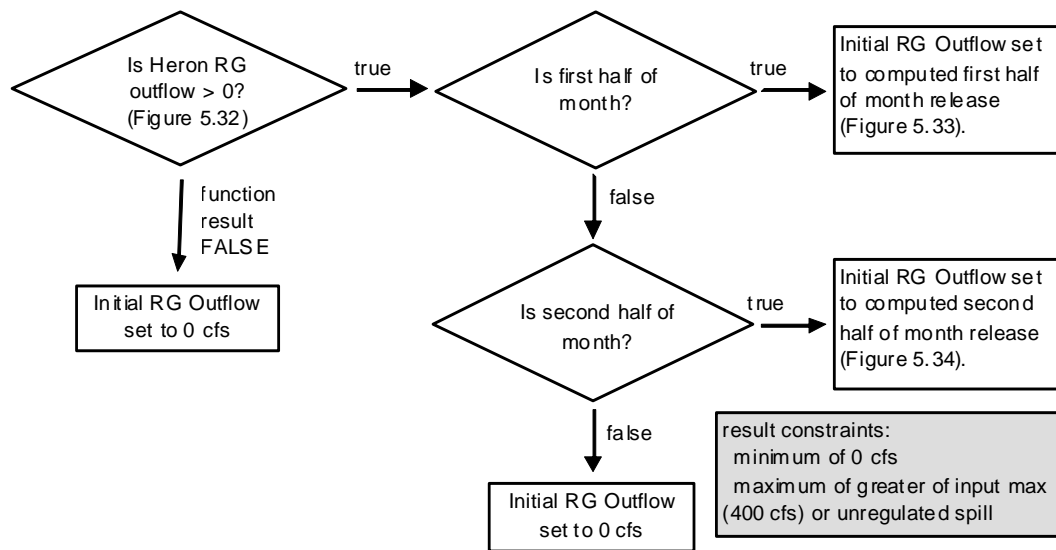


Figure A.31. Flow Chart for Initial Checks to Set Rio Grande Release from Heron

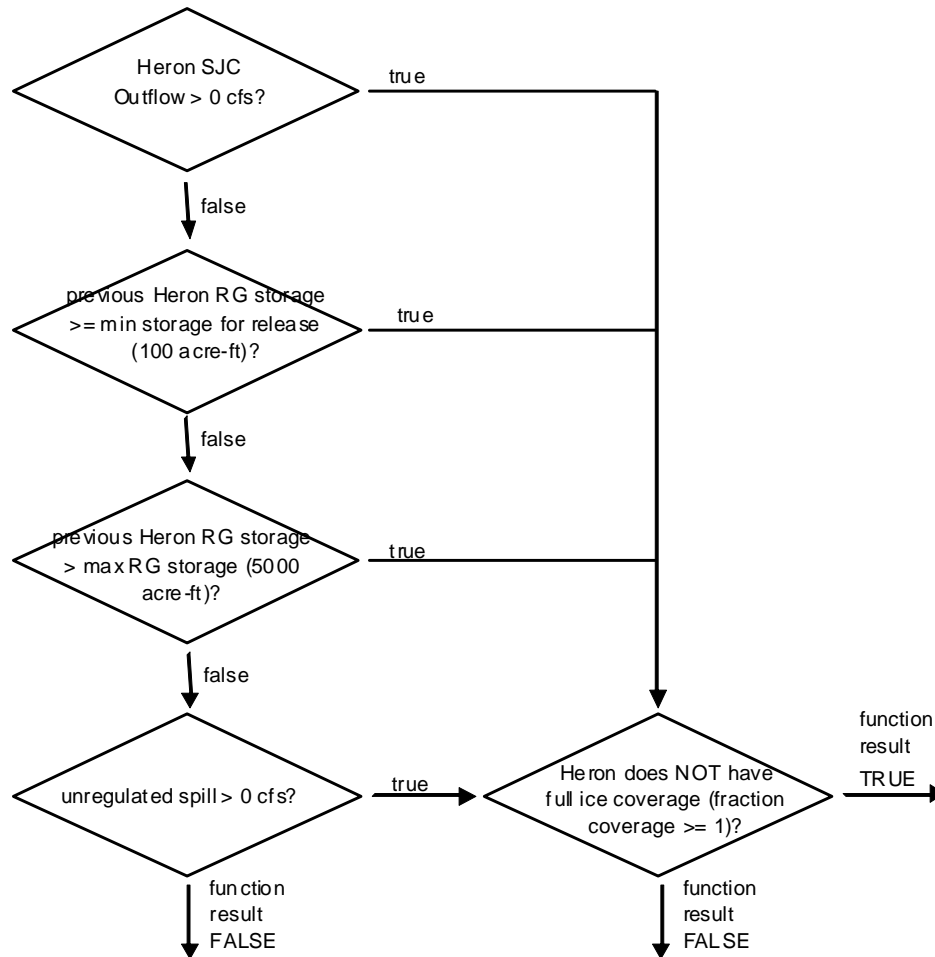


Figure A.32. Flow Chart Depicting Details of Check of Heron Rio Grande Outflow > 0

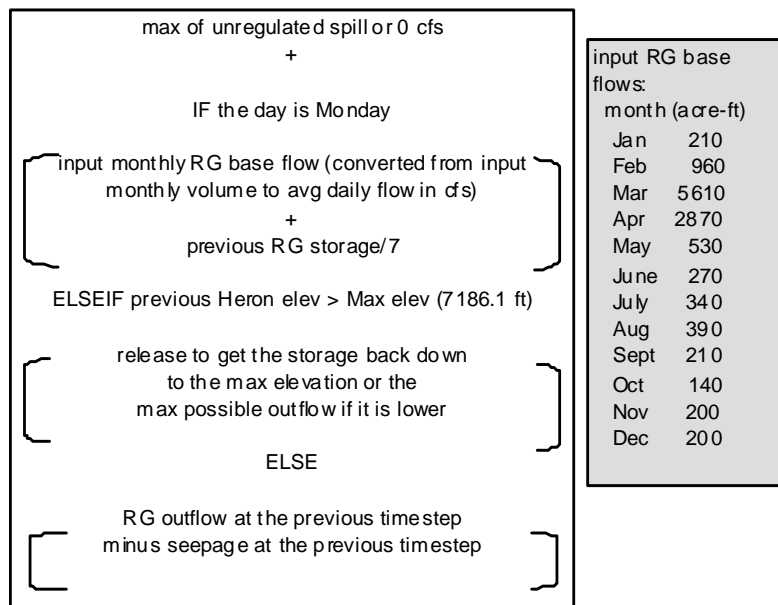


Figure A.33. Calculation for Rio Grande Release – First Half of Month

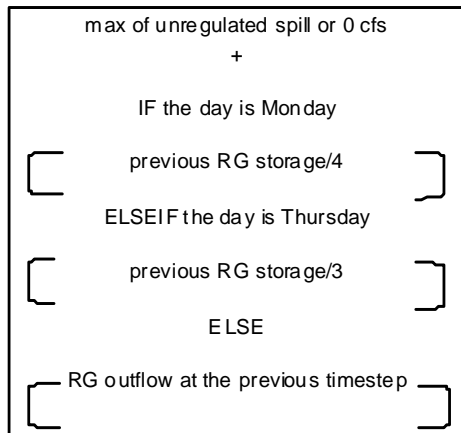


Figure A.34. Calculation for Rio Grande Release – Second Half of Month

### A.9.12. ElVadoIndianAccountTransfer

*Explanation:* This rule computes the value for the accounting supply to transfer water in the Indian Storage account that is not needed based on updated calculations for the Indian storage requirement over to the Rio Grande account. The value is determined as a function of the difference in the storage in the Indian storage account at El Vado Reservoir and the updated Indian storage requirement.

*Rule Execution:* No execution constraints included with this rule.

*Rule Logic:* This function references the user-defined IndianStorageTransferToRioGrande function in the ElVado Account Functions utility group. This function includes an IF THEN ELSE statement to see if the Indian storage requirement is less than the previous Indian storage requirement and the storage in the Indian storage account at El Vado Reservoir for the previous timestep is greater than the Indian storage requirement. If so, the value for the supply for the IndianStorageElVadoToRioGrandeElVado account is set to the storage in the Indian storage account at El Vado Reservoir for the previous timestep minus the Indian storage requirement all divided by the number of days remaining in the current month, or remaining in the simulation if before the end of the month. Otherwise, the supply is set to 0.0 cfs.

The previous Indian storage requirement is referenced using the user-defined GetIndianStorageRequirement function in the Get Data functions utility group. The Indian storage requirement is computed using the user-defined IndianStorageRequirement function in the Get Data Functions utility group. This function computes the requirement as the maximum of zero or the value for the IndianStorageReq time series slot in the ElVadoData data object referenced with the user-defined GetIndianStorageRequirement function minus the sum of the supplies since the start of the current month, or the start timestep if it is later, for the



IndianStorageElVadoToRioGrandeBlwElVado account as computed using the user-defined SumMonthlyIndianCallToDate function in the Account Summing Functions utility group. If the end of the current month is after the finish timestep, the same equation is used to compute the Indian storage requirement but the Indian storage requirement for the finish timestep, as opposed to the end of the current month, is used and is interpolated using the finish timestep and the values in the IndianStorageReq time series slot in the ElVadoData data object for the end of the current month and the end of the previous month. The interpolation is completed using the user-defined InterpolateIndianStorageReq function in the Get Data Functions utility group.

### **A.9.13. SetElVadoIndianStorageInflow**

*Explanation:* This rule is used to set the supply to transfer water from the Rio Grande account to the Indian storage account to satisfy the Indian storage requirement. If the stipulations of Article VII of the Compact are in effect, the transfer is set to the current Rio Grande inflow minus the minimum Rio Grande release. If the stipulations of Article VII of the Compact are not in effect, the transfer is set to the total amount of Rio Grande water that will be stored until the Indian storage requirement is satisfied. Effectively, all available Rio Grande flow is designated for P&P storage until the Indian storage requirement is met.

*Rule Execution:* If the value for the supply for the RioGrandeElVadoToIndianStorageElVado account for the current timestep is a NaN, the rule fires. This ensures that the rule only fires once per timestep.

*Rule Logic:* This rule references the user-defined IndianStorageInflow function in the ElVado IndianStorage Account Functions utility group. This function includes an IF THEN ELSE statement to see if the stipulations of Article VII of the Compact are in effect as determined with the user-defined CompactVIIinEffect function. If so, the supply is set to the minimum of the result from the user-defined ComputeIndianStorageSpace function or the Rio Grande inflow to El Vado Reservoir minus the minimum Rio Grande inflow to El Vado Reservoir. If the stipulations of Article VII of the Compact are not in effect, the supply is set to the minimum of the result from the user-defined ComputeIndianStorageSpace function or the Rio Grande inflow to El Vado Reservoir minus the result from the user-defined ElVadoRGReleaseForIndianStorageInflow function in the ElVado Account Functions utility group.

The ComputeIndianStorageSpace function computes a flowrate as the maximum of zero or the current Indian storage requirement minus the Storage and GainLoss values in the Indian storage account at El Vado Reservoir. The predefined VolumeToFlow function is used to convert the storage values to a daily flow.

The current Rio Grande inflow to El Vado Reservoir is computed with the user-defined CurrentRGInflow function in the General Compute Functions utility group as the value

for the RGOutflow time series slot in the HeronData data object referenced using the user-defined RGOutflow function plus the value for the Local Inflow time series slot in the ElVadoLocalInflow reach object plus the value for the Seepage time series slot in the Heron storage reservoir object if the value is not a NaN. The minimum Rio Grande inflow to El Vado Reservoir is input to the MinRGOutflow periodic slot in the ElVadoData data object and referenced using the user-defined MinRGOutflow function in the Get Data Functions utility group and restricted to be no larger than the Rio Grande inflow to El Vado Reservoir.

The ElVadoRGReleaseForIndianStorageInflow function adds the input value for the Texas Call time series slot in the RioGrandeCompact data object for the current timestep referenced using the user-defined TexasCall function in the Get Data Functions utility group to the maximum of the following values: the result from the user-defined ComputeElVadoTargetRelease, the minimum Rio Grande outflow from El Vado Dam, or sum of the result from the user-defined MinComputedElVadoMRGCDDReleaseForIndianStorageInflow function, the value in the Indian Call time series slot in the Indian data object for the timestep equal to the current timestep plus two timesteps, and the result from the user-defined ElVadoESAReserveRG function.

#### **A.9.14. SetInflowToMRGCDDroughtStorage**

*Explanation:* If New Mexico has relinquished Compact credits with subsequent allocations for storage of Emergency Drought water for MRGCD, this rule allows for a portion of the Rio Grande inflow above the minimum Rio Grande release to be stored in the MRGCD Drought account for meeting that storage allocation for Emergency Drought water for MRGCD when the stipulations of Article VII of the Compact are in effect.

*Rule Execution:* If the value for the RioGrandeElVadoToMRGCDDroughtElVado supply is a NaN, the rule fires.

*Rule Logic:* The user-defined MRGCDDroughtInflow function in the ElVado Account Functions utility group is referenced in this rule. This function includes an IF THEN ELSE statement to see if the value in the storage in the MRGCDDrought account is less than the allocation as checked with the user-defined HasMaxAccountStorageNOTBeenReached function in the RelinquishedNMCreditFunctions utility group and whether the stipulations of Article VII of the Compact are in effect (Refer to the SetCompactVIISwitch Rule) as determined with the user-defined CompactVIIinEffect function. If not, the value for the RioGrandeElVadoToMRGCDDroughtElVado supply for the current timestep is set to zero. If so, the supply is set to the result from the user-defined CurrentRGInflow function minus the result from the user-defined IndianStorageInflow function minus the result from the user-defined MinRGOutflow function (or the value for the ComputedMinMRGCDDRelease if higher) plus the result from the IndianStorageTransferToRioGrande function (plus an adjustment for a negative storage

in the Rio Grande account if relevant) all multiplied by the quantity of the result for the user-defined ComputeMRGCDDroughtSpace function divided by the quantity of the result from the user-defined ComputeSupplementalESASpace function plus the result from the user-defined ComputeMRGCDDroughtSpace function. The predefined MinItem function is used to restrict the result from this function to the result from the user-defined ComputeMRGCDDroughtSpace function.

The user-defined ComputeMRGCDDroughtSpace function in the ElVado Account Functions utility group includes an IF THEN ELSE statement to see if the result from the user-defined HasMaxAccountStorageNOTBeenReached function is true. If not, the result is 0.0 cfs. If so, the calculation is completed as the value in the MRGCDDroughtAllocation series slot in the ElVadoData data object minus the Storage in the MRGCDDrought storage account at El Vado Reservoir for the previous timestep minus the Accrual in the MRGCDDrought storage account at El Vado Reservoir for the previous timestep. The equation in the user-defined ComputeSupplementalESASpace function in the ElVado Account Functions utility group is the same as the equation in the ComputeMRGCDDroughtSpace function except the corresponding values for the SupplementalESA storage account are referenced instead.

#### **A.9.15. SetInflowToSupplementalESAStorage**

*Explanation:* If New Mexico has relinquished Compact credits with subsequent allocations for storage of Emergency Drought water for ESA purposes, this rule allows for a portion of the Rio Grande inflow above the minimum Rio Grande release to be stored in the SupplementalESA account for meeting that storage allocation for Emergency Drought water for ESA purposes when the stipulations of Article VII of the Compact are in effect.

*Rule Execution:* If the value for the RioGrandeElVadoToSupplementalESAElVado supply is a NaN, the rule fires.

*Rule Logic:* The user-defined SupplementalESAInflow function in the ElVado Account Functions utility group is used to set the value for the RioGrandeElVadoToSupplementalESAElVado supply for the current timestep. The function computes the result using the same equation used for the SetInflowToMRGCDDroughtStorage Rule except the ratio of the inflow to be transferred to the SupplementalESA account is based on the result from the user-defined ComputeSupplementalESASpace function as opposed to the ComputeMRGCDDroughtSpace function.

#### **A.9.16. ComputeElVadoRGRelease**

*Explanation:* This rule computes the Rio Grande release from El Vado Dam. If the stipulations of Article VII of the Compact are in effect, the release is set to bypass

inflows but subtracting any transfers to the MRGCD Drought, Supplemental ESA, and the Indian storage accounts at El Vado Reservoir from the Rio Grande inflow. Any Rio Grande releases from these accounts are added if relevant and the transfer of water from the Indian storage account to the Rio Grande account is added. The result is checked against the minimum Rio Grande outflow. If the stipulations of Article VII are not in effect, the release is set to fill the reservoir by a fill date while meeting downstream demands for MRGCD, Indian, and ESA. The reservoir is filled with reference to the computed percent Rio Grande release as computed with the SetPercentRGRelease rule.

*Rule Execution:* If the value for the RGOutflow time series slot in the ElVadoData data object for the current timestep is a NaN, the rule fires.

*Rule Logic:* This rule references the user-defined ElVadoRGRelease function in the ElVado Account Functions utility group. If the stipulations of Article VII of the Compact are in effect, the outflow is computed as the result from the user-defined CurrentRGInflow function for El Vado Reservoir minus the result from the user-defined MRGCDDroughtInflow function minus the result from the user-defined SupplementalESAInflow function minus the result for the user-defined IndianStorageInflow function plus the result for the user-defined ElVadoMRGCDReleaseRG function plus the result from the user-defined ElVadoESAResaleRG function plus the result from the user-defined ElVadoIndianCall function plus the result from the user-defined IndianStorageTransferRelease function. The result is restricted to the result from the MinRGOutflow function using the predefined Max function. If the stipulations of Article VII are not in effect, the release is set to the result from the user-defined ElVadoRGSeasonalRelease function in the ElVado Account Functions utility group.

The user-defined ElVadoMRGCDReleaseRG function is in the Get Data Functions utility group. The ElVadoESAResaleRG and the IndianStorageTransferToRioGrande functions are in the ElVado Account Functions utility group, and the user-defined ElVadoIndianCall function is in the Indian Storage Functions utility group.

The ElVadoRGSeasonalRelease function computes the outflow as the result from the user-defined TexasCall function plus the maximum of either the result from the user-defined ComputeElVadoTargetRelease function, the user-defined MinRGOutflow function for El Vado Reservoir, or the result from the MinComputedElVadoMRGCDRelease plus the result from the user-defined ElVadoESAResaleRG function plus the value for the Indian Call time series slot in the Indian data object for the timestep equal to the current timestep plus two timesteps, or the finish timestep if it is earlier. A flowchart for that depicts the logic for setting the release of Rio Grande water from El Vado Dam is presented in Figures A.35 that includes the step for computing a percent release as calculated with the SetPercentRGRelease rule.

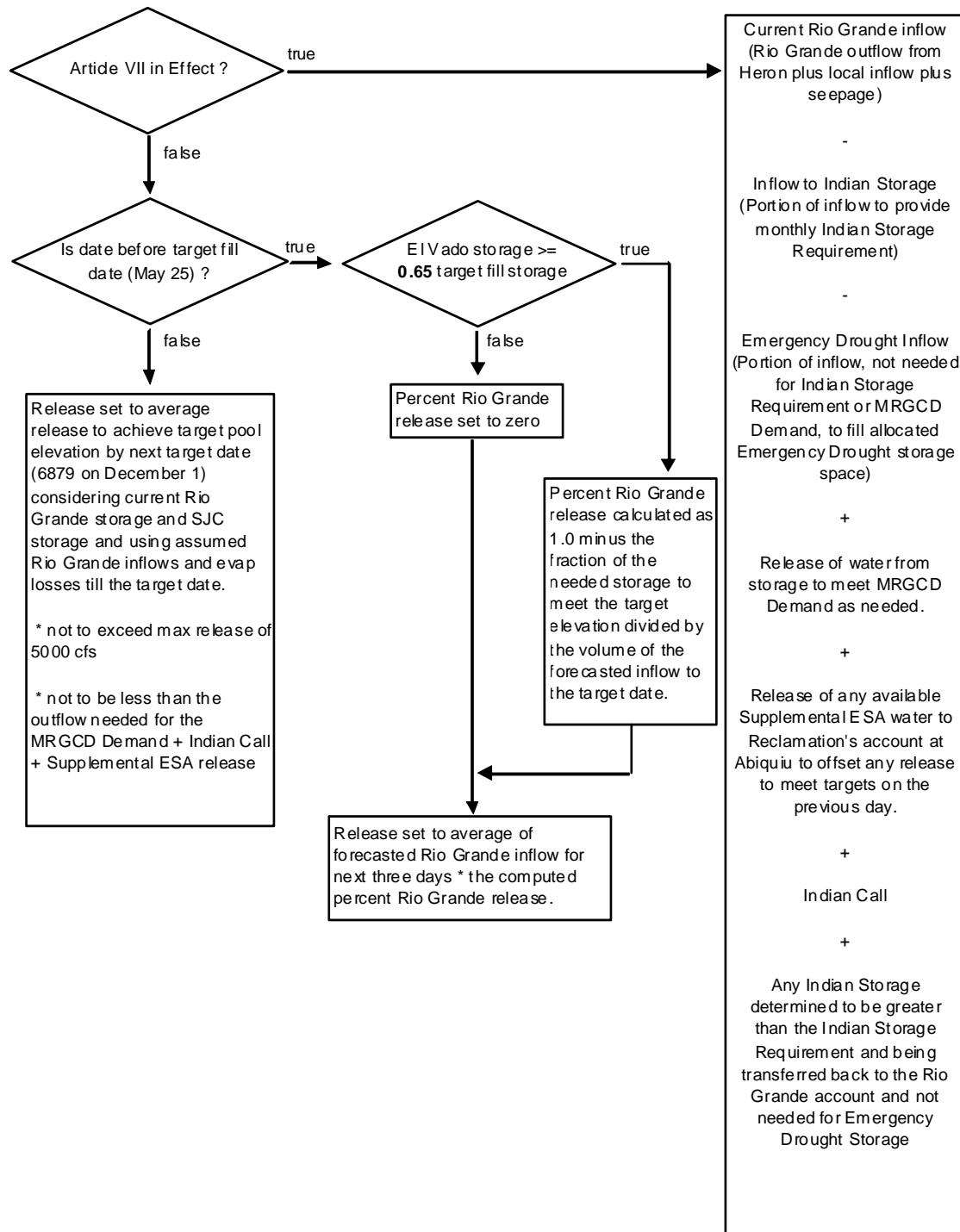


Figure A.35. Flow Chart with Logic to Set Release of Rio Grande Water from El Vado

### A.9.17 EstimateHeronSJRelease

*Explanation:* An initial total release of San Juan-Chama Project water from Heron Reservoir is computed with this rule. The calculation includes releases for different purposes including to fill downstream allocated storage space, for contractors letter water deliveries to payback the river at Otowi, and for the Cochiti rec pool.

*Rule Execution:* This rule fires if the rule has not fired already for the current timestep as checked with the predefined HasRuleFiredSuccesfully function.

*Rule Logic:* The user-defined ComputeHeronSJOutflow function in the Heron Account Functions utility group computes the San Juan outflow as the result from the user-defined ElVadoDeliveryRelease function plus the maximum of either the result from the user-defined MinHeronSJOutflow function or the volume of the payback debt associated with accounting supplies that have a release type of AlbuquerquePaybacks and a destination of Albuquerque\_Abiquiu plus the sum of the results from the following user-defined functions: HeronAccountFillRelease, HeronOtowiAndJemezPaybackRelease, MRGCDPaybackRelease, HeronCochitiRecPoolRelease, and HeronWaiverRelease. These functions compute the release to meet different purposes. Refer to Figure A.36 for a diagram that illustrates the different aspects of the calculation.

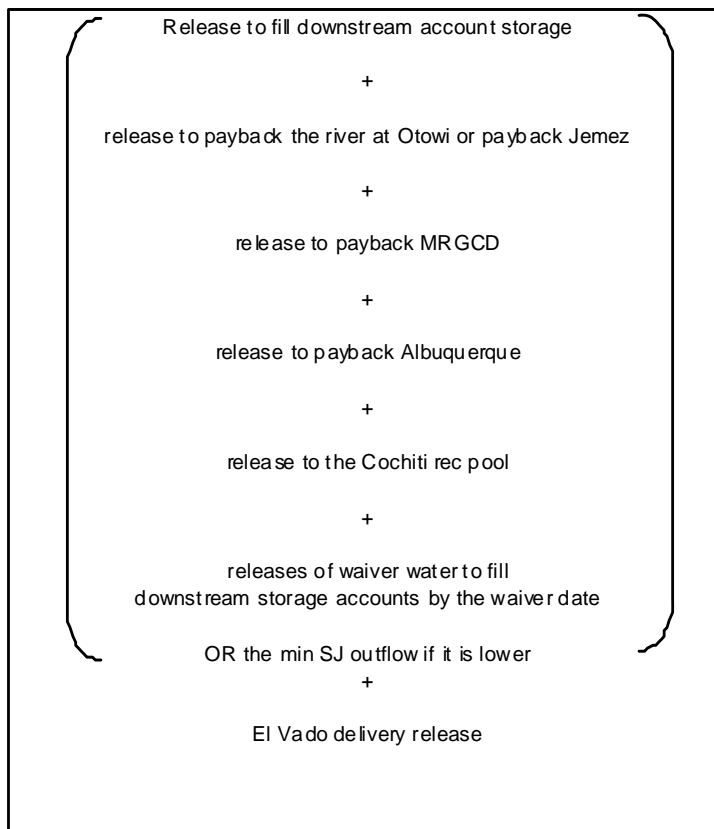


Figure A.36. Components of Initial Total Release of SJC Water from Heron Reservoir

### **A.9.18. SetCarryover**

*Explanation:* On January 1 of each year, this rule sets the value for the Carry Over for all storage accounts at Heron Reservoir that have downstream storage accounts at El Vado and/or Abiquiu Reservoirs if waivers are in effect for the previous year. The carryover is set to the amount of water in the storage account at Heron Reservoir on December 31. This is a step needed for maintaining waiver water in the storage accounts for contractors for San Juan-Chama Project water.

*Rule Execution:* If the timestep is January 1, the rule fires.

*Rule Logic:* A FOREACH loop is used to set the value for the Carry Over for all the accounts at Heron Reservoir that have San Juan storage accounts at El Vado Reservoir and/or Abiquiu Reservoir. The user-defined StringListsUnion function in the General List Function utility group is used to create the list of accounts. If the value for the WaiverSwitch time series slot in the HeronData data object for the previous year is equal to 1.0 as identified with the user-defined PreviousYearIsWaiverYear function in the General Boolean Functions utility group which references the user-defined PreviousYearWaiverSwitch function in the Get Data Functions utility group, the Carry Over is set to the Storage for the account at Lake Heron for the previous timestep as referenced with the user-defined PreviousAccountStorage function. Otherwise, the Carry Over is set to zero.

A second FOREACH loop is used to set the value for the Carry Over for all other accounts at Heron Reservoir to zero. The user-defined StringListsNonIntersect function in the General List Functions utility group is used with the user-defined StringListsUnion function to create the list of storage accounts at Heron Reservoir for San Juan water that do NOT have San Juan storage accounts at El Vado Reservoir or Abiquiu Reservoir. Refer to Figure A.37 for a flowchart portraying the logic used for this rule.

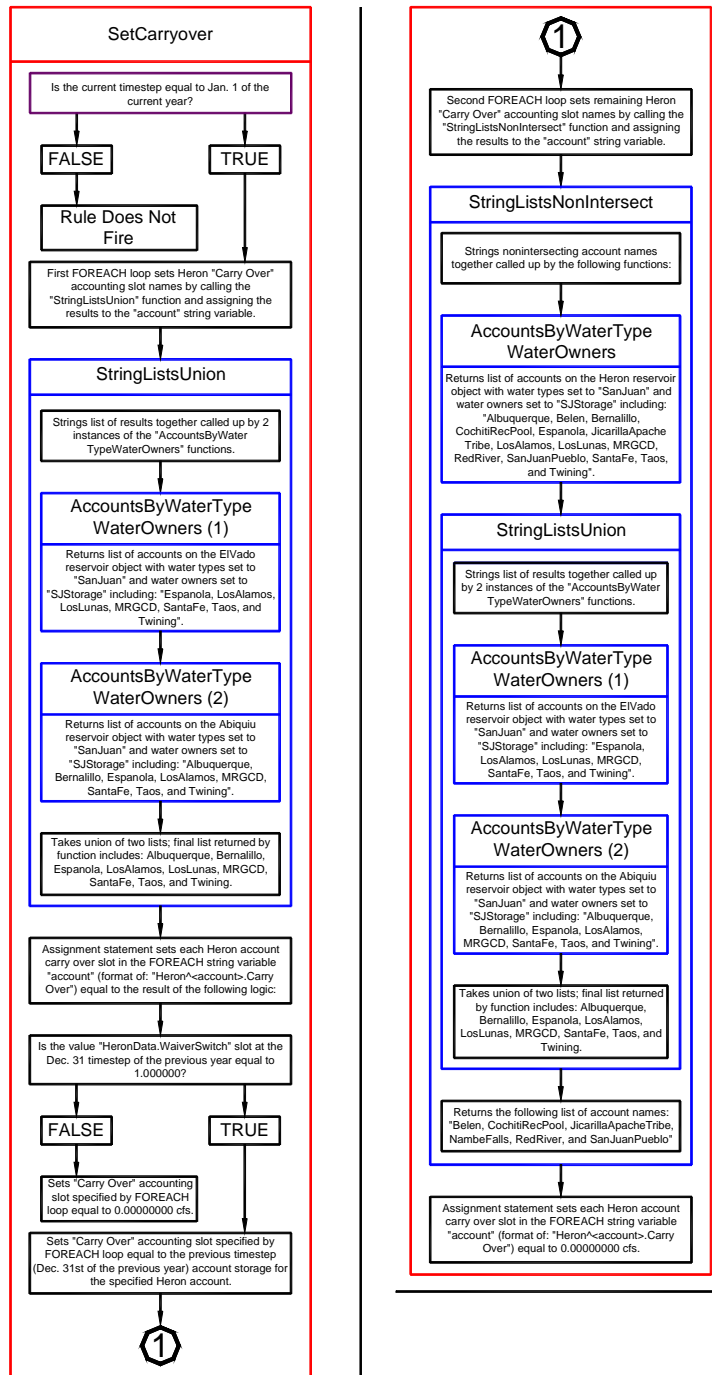


Figure A.37. Flowchart for the SetCarryover Rule



## **A.10. San Juan Diversions**

The rules in this policy group are used for setting the San Juan diversions with consideration for the available flow in the tributaries to the San Juan River, all diversion and tunnel capacities, and legal constraints for the Project diversions. Diversions result in modeled flows into Heron Reservoir via the Azotea tunnel.

### **A.10.1. ComputeAnnualSJDiversions**

*Explanation:* On January 1 of each year, this rule sums the annual diversion of San Juan water over the previous year. The values are then referenced when checking annual and 10-year restrictions on San Juan-Chama Project diversions when setting diversions.

*Rule Execution:* If the current timestep is January 1 and not the Start Timestep and the value for the AnnualDiversions time series slot in the SanJuanChamaDiversions data object for December 31 of the previous year is a NaN, the rule fires.

*Rule Logic:* The value for the AnnualDiversions time series slot in the SanJuanChamaDiversions data object for the previous timestep is set to the summation of the values in the Total time series slot in the SanJuanChamaDiversions data object for the January 1 of the previous year, or the start timestep if it is later, through the previous timestep. The predefined SumSlotOverTime function is used to complete the summation and the predefined FlowToVolume function is used to convert the flows to a volume.

### **A.10.2. San Juan Diversions**

*Explanation:* The San Juan Diversions at the Blanco, Little Oso, and Oso diversions are determined with this rule. If an input maintenance switch is checked for a diversion, an input diversion will be referenced for the corresponding diversion. Diversions will be limited if necessary based on the annual limit, limited diversion for a decade, or the available space at Lake Heron. If the Project diversion is limited, separate calculations are used to determine the diversions. Otherwise, the diversions are set to the input diversion capacity. These rules capture detailed aspects of diversion and tunnel capacities that may ultimately affect the total Project diversion volume to Heron Reservoir via the Azotea tunnel.

*Rule Execution:* There are no execution constraints for this rule.

*Rule Logic:* A FOREACH loop is used to set the Diversion Request values for the current timestep for all the diversions in the San Juan Diversions subbasin. An exterior IF THEN ELSE statement is used to identify whether the maintenance switch is on for the diversion as identified with the user-defined IsMaintenanceSwitchOn function in the Maintenance Functions utility group. If so, the Diversion Request is set to the input value in the

MaintenanceFlow table slot for the data object for the corresponding diversion. An interior IF THEN ELSE statement is used to identify whether the diversion is limited based on the annual diversion limit, decade diversion limit, or the space in Lake Heron using the user-defined DiversionIsLimited function in the San Juan Diversion Functions utility group. If not, the Diversion Request is set to the input value in the Max Diversion table slot for the corresponding diversion. If so, the limited remaining diversion for the year, decade, or based on the available space in Heron Reservoir is converted to a flow rate for one timestep and compared to the value in the SmallDivThreshold table slot in the SanJuanChamaRule data object. If the threshold is not exceeded, the Diversion Request is set with the user-defined SmallDivCalc function in the San Juan Diversion Functions utility group. Otherwise, the value is set using the user-defined LargeDivCalc function in the San Juan Diversion Functions utility group.

This rule also includes two additional assignment statements to set the values in the Diversion Request time series slots in the OsoTunnelInlet and AzoteaTunnelInlet diversion objects to the value in the Max Diversion table slot for the corresponding diversion as referenced using the user-defined Capacity function in the San Juan Diversion Functions utility group. Refer to Figures A.38 for a flowchart that depicts the logic used for setting the diversions.

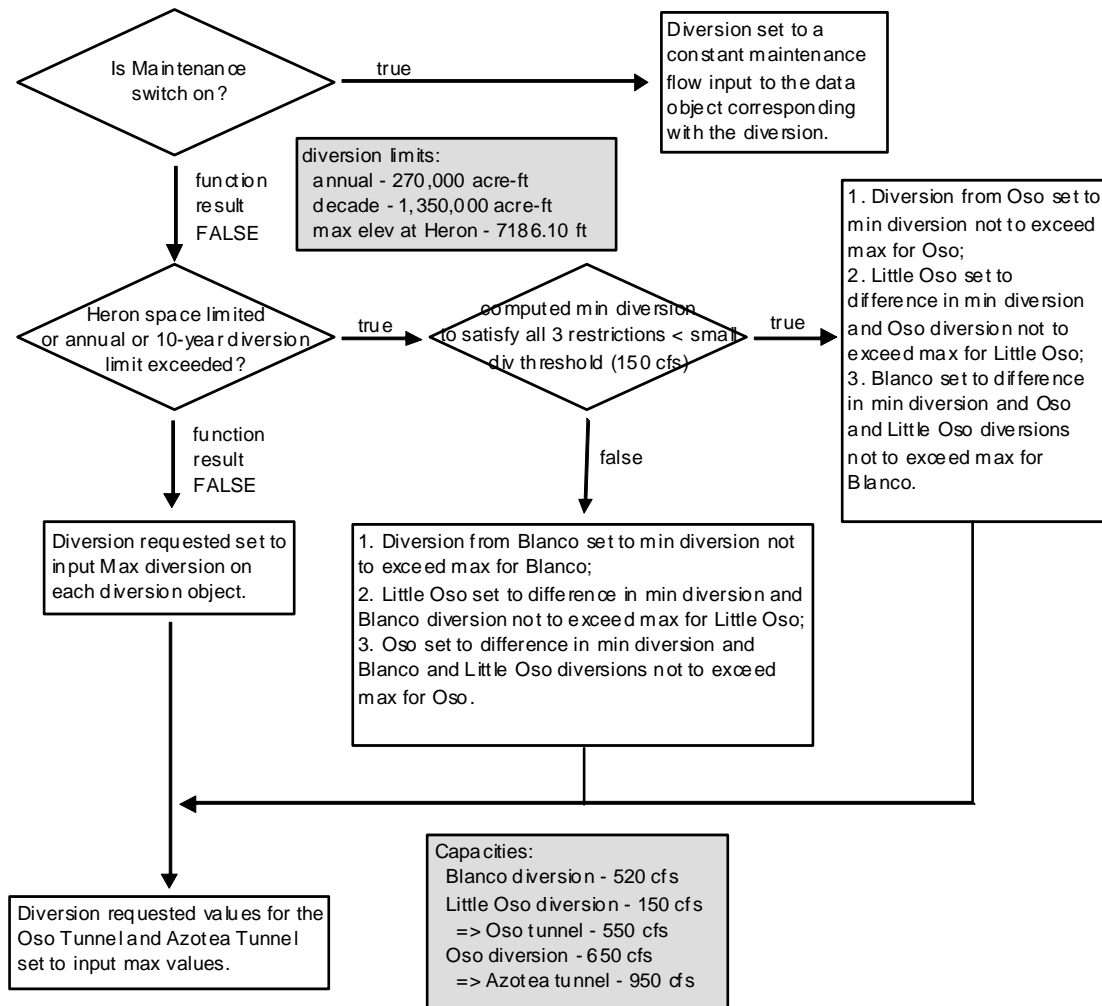


Figure A.38. Flow Chart Depicting Logic for Setting San Juan Diversions

## **A.11. Heron**

The rules in this policy group are used to ultimately determine the total outflow from Heron Dam and include checks against the reservoir ice coverage, whether the reservoir is spilling, and the maximum daily pool elevation change of 1 foot.

### **A.11.1. HeronSJReleaseRestrictions**

*Explanation:* This rule checks to see if Heron Reservoir has full ice coverage or if it must spill. If so, the release of San Juan-Chama Project water is reset to 0.0 cfs. The ice coverage is computed as part of the CurrentSurfaceAreaPanAndIce evaporation method.

*Rule Execution:* This rule fires if it has not fired yet as determined with a reference to the predefined HasRuleFiredSuccessfully function.

*Rule Logic:* This rule contains an IF THEN statement to set the value of the SJOutflow time series slot in the HeronData data object for the current timestep to 0.0 cfs if the result from either the user-defined HeronHasFullIceCover or HeronMustSpill functions in the Heron Functions utility group is true. The former function checks to see if the input value in the Surface Ice Coverage time series slot in the Heron storage reservoir object for the current timestep is greater than or equal to the value in the IceCoverageThreshold table slot in the HeronData data object. The latter function first computes the storage of water above the maximum pool elevation as the value for the Storage time series slot in the Heron storage reservoir object for the previous timestep minus the maximum storage that corresponds to the MaxElevation input in the ReservoirData table slot in the HeronData data object. If that value plus the value in the Inflow time series slot in the Heron storage reservoir object minus the value in the Inflow1 time series slot in the AzoteaWillow confluence object is greater than the maximum outflow as determined using the predefined GetMaxOutflowGivenHW function, the result for the function is TRUE. Refer to Figure A.39 for a flowchart that depicts the logic used in this rule.

### **A.11.2. HeronOutflow**

*Explanation:* Unless a total outflow for Heron Dam has been input for the current timestep, the value is set to the sum of the separate initial computed values for the release of native Rio Grande water and the release of San Juan-Chama Project water if the resulting total release is physically legitimate with consideration for the outlet works. If not, the release is reset to reflect the restriction of the Heron Dam outlet works.

*Rule Execution:* If the value for the Outflow time series slot in the Heron storage reservoir object for the current timestep is a NaN, the rule fires.

*Rule Logic:* This rule contains an IF THEN ELSE statement to see if the value in the TotalOutflow time series slot in the HeronData data object for the current timestep as referenced using the user-defined TotalOutflow function in the Get Data Functions utility group is a NaN. If not, the value for the Outflow time series slot in the Heron storage reservoir object for the current timestep is set to that TotalOutflow value. Otherwise, the Outflow is set to the sum of the RGOutflow and SJOutflow for Heron Dam checked with the user-defined CheckThisResPhysicalConstraints function (Section A.11.2.1) in the General Compute Functions utility group with. Refer to Figure A.39 for a flowchart that depicts the logic used in this rule.

#### A.11.2.1. CheckThisResPhysicalConstraints

*Explanation:* This function checks a predetermined outflow against the physical constraints of the outlet works for a dam. First, if there would be an unregulated spill, the outflow is reset to that spill if it is greater than the predetermined outflow. Otherwise, the predetermined outflow is checked to assure it does not exceed the maximum outflow that can be released through the outlet works or the release that would reduce the storage down to the invert level of the outlet works (or the level corresponding to a maximum release of 0.0 cfs).

*Function Logic:* This function includes a WITH DO loop to determine the minimum spill for the current reservoir situation using the predefined GetMinSpillGivenInflowRelease function. An IF THEN ELSE statement is used to see if the predetermined outflow is less than the determined minimum spill. If so, the outflow is reset to the minimum spill. If not, the function result is set to the minimum of the predetermined outflow or the result from the user-defined ComputeMaxOutflow function in the General Compute Functions utility group which references the predefined GetMaxOutflowGivenInflow function or the result from the user-defined ComputeOutflowAtGivenStorage function in the General Compute Functions utility group which references the predefined SolveOutflow function to solve for the outflow that would reduce the storage down to the elevation corresponding with 0 cfs in the Max Release table in the storage reservoir object.

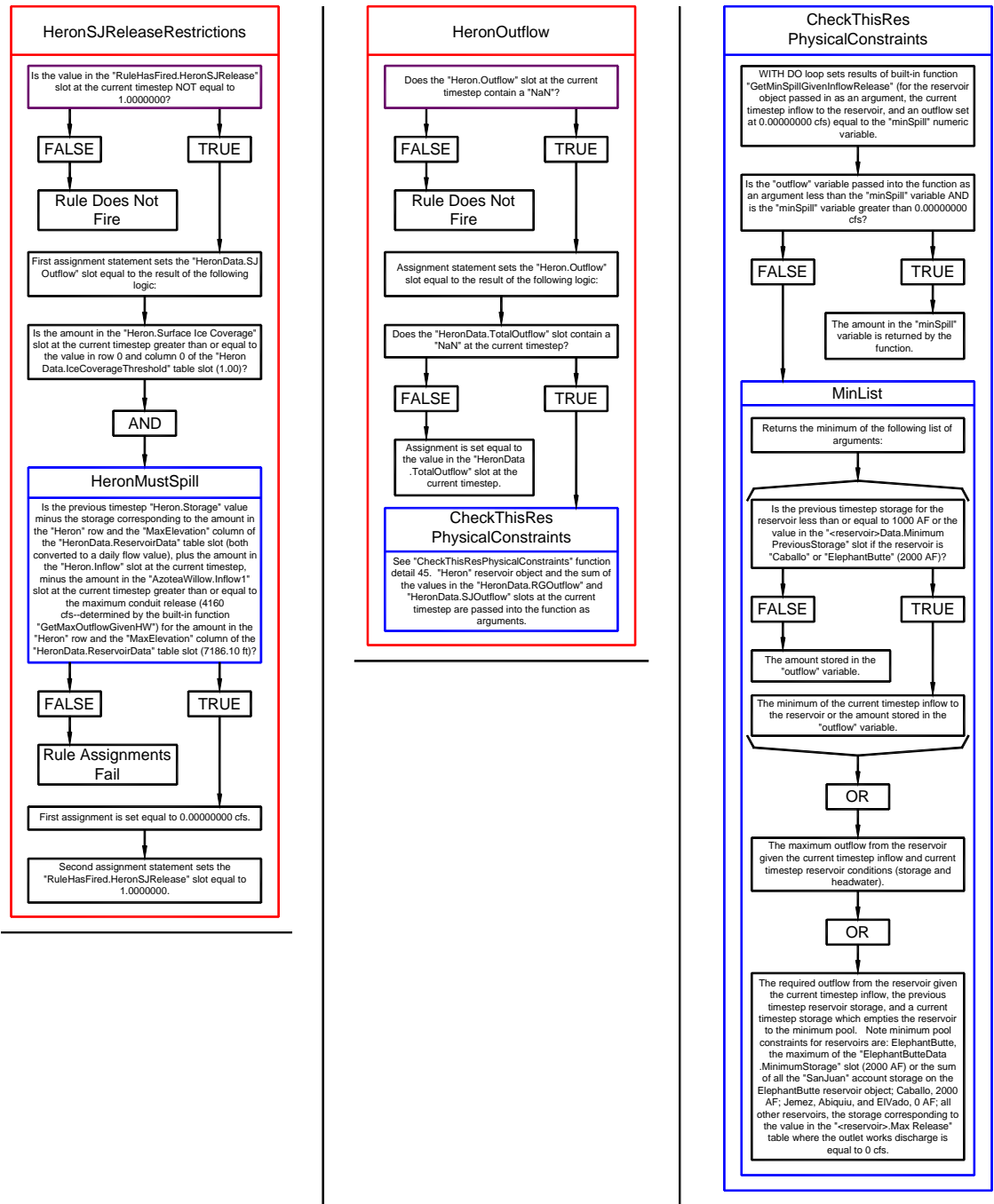


Figure A.39. Flowcharts for the HeronSJReleaseRestrictions and HeronOutflow Rules

### **A.11.3. HeronCheckDeltaStorage**

*Explanation:* If the change to the pool elevation, resulting from the initial set outflow for the current timestep, is greater than the limited change of 1 foot, the outflow is reset to restrict the change to the pool elevation to the input value of 1 foot.

*Rule Execution:* If the change in the pool elevation as determined with the user-defined ComputeDeltaPoolElevWOAbs function in the General Compute Functions utility group is greater than the value input to the maxDeltaPoolElev table slot in the HeronData data object and the value in either the RGOutflow or the SJOutflow time series slot in the HeronData data object for the current timestep is greater than zero, the rule fires.

*Rule Logic:* This rule references the user-defined HeronLimitDeltaStorage function in the Heron Functions utility group to compute the outflow corresponding to a decrease in the pool elevation equal to the input value in the maxDeltaPoolElev table slot in the HeronData data object. The predefined ElevationToStorage and VolumeToFlow functions are referenced in this function. The value in the Outflow time series slot in the Heron storage reservoir object is then set to the result after it is checked with the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1 for more information on this function). This rule also includes a PRINT statement to print a comment to the Diagnostics Output window that the change in storage was limited. Refer to Figure A.40 for a flowchart illustrating the logic in this rule.

### **A.11.4. SetHeronRGAccount**

*Explanation:* This rule sets the supply for the release from Rio Grande account at Heron Reservoir based on the determined outflow. If the Rio Grande outflow is greater than the Heron outflow, the supply is set to the Heron outflow. If the Heron outflow is greater than the sum of the Rio Grande and San Juan outflow, the reconciled release is set to the Heron outflow minus the San Juan outflow, or the minimum Rio Grande outflow if it is higher. Otherwise, the supply is set to the Rio Grande outflow.

*Rule Execution:* If the value for the Supply at Heron Reservoir with a release type of Rio Grande and a destination of RioGrande is a NaN, the rule fires. The GET STRING @INDEX command is used with the user-defined SuppliesByReleaseTypeDestination function in the Account Lists utility group to identify the supply name.

*Rule Logic:* The value for the Supply at Heron Reservoir with a release type of Rio Grande and a destination of RioGrande for the current timestep is set to the result from the user-defined ReconcileRGOutflow function in the General Account Functions utility group. For Heron, this ReconcileRGOutflow function includes an IF THEN ELSE statement to see if the value for the RGOutflow time series slot in the HeronData data object for the current timestep as referenced using the user-defined RGOutflow function in the General Account Functions utility group is greater than the value in the Outflow

time series slot in the Heron storage reservoir object. If so, the function result is set to that Outflow for the Heron storage reservoir object. Otherwise, a second IF THEN ELSE statement is used to see if the value for that Outflow is greater than the sum of the values in the RGOutflow and SJOutflow time series slots in the HeronData data object. If so, the function result is set to the maximum of either the minimum Rio Grande outflow (0.0 cfs for Heron Dam) or the difference between the value in the Outflow time series slot in the Heron storage reservoir object and the value in the SJOutflow time series slot in the HeronData data Object. If not, the function result is set to the value in the RGOutflow time series slot in the HeronData data object for the current timestep. A flowchart for this rule is presented in Figures A.40 and A.41.

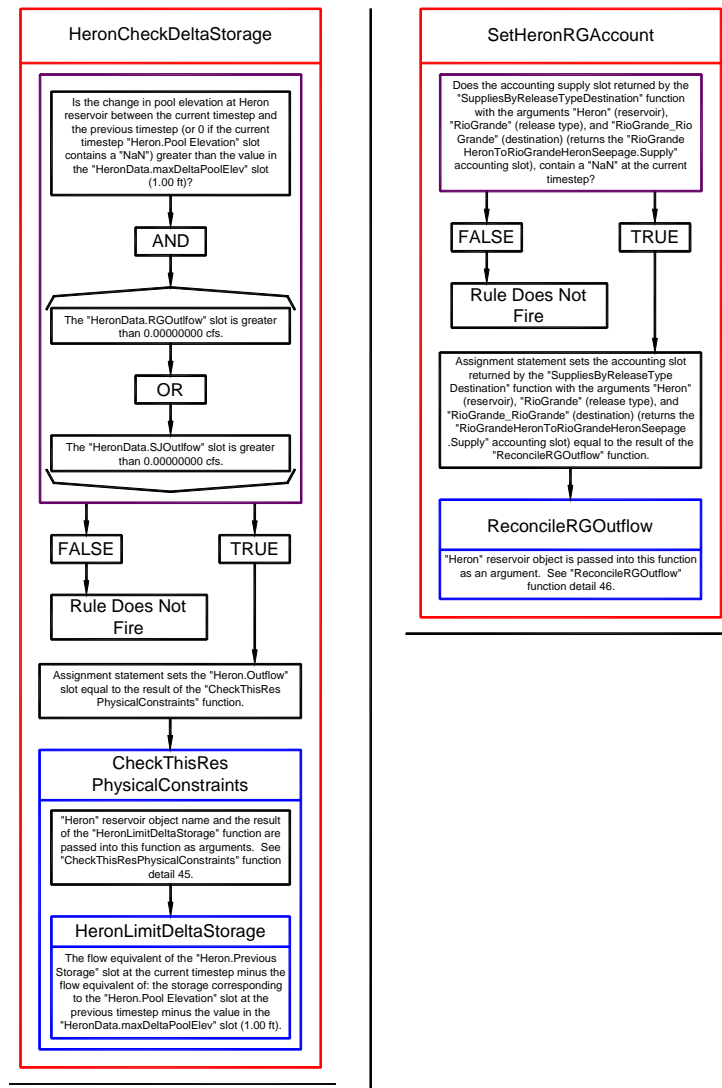


Figure A.40. Flowcharts for the HeronCheckDeltaStorage and SetHeronRGAccount Rules



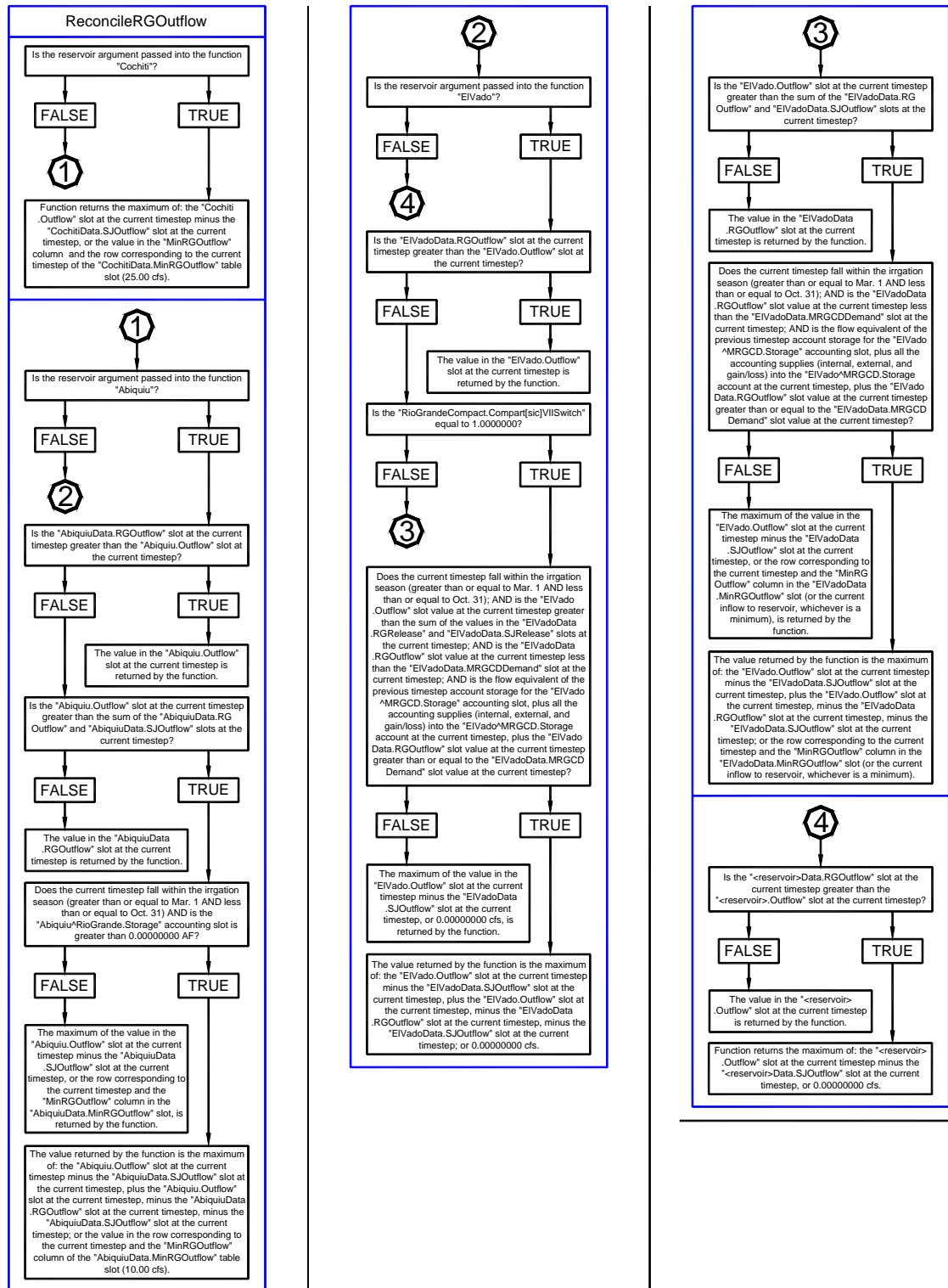


Figure A.41. Flowchart for the ReconcileRGOutflow Function

### A.11.5. CheckHeronRGOutflow

*Explanation:* This rule aborts the simulation if the supply for the release of Rio Grande water from the account at Heron Reservoir was not set. This rule helps with debugging because a simulation will stop if the Rio Grande supply was not set and the problem and any missing input can be identified by backtracking in the calculations from the point in the simulation that the run aborted.

*Rule Execution:* There are no execution constraints for this rule.

*Rule Logic:* If the value for the Supply at Heron Reservoir Lake with a release type of Rio Grande and destination of RioGrande is a NaN, the STOP\_RUN command is used to stop the simulation, and the PRINT statement is used to print a comment to the Diagnostics Output window that the run stopped because the account slot value was not set. A screen capture of the RiverWare RPL for this rule is presented in Figure A.42.

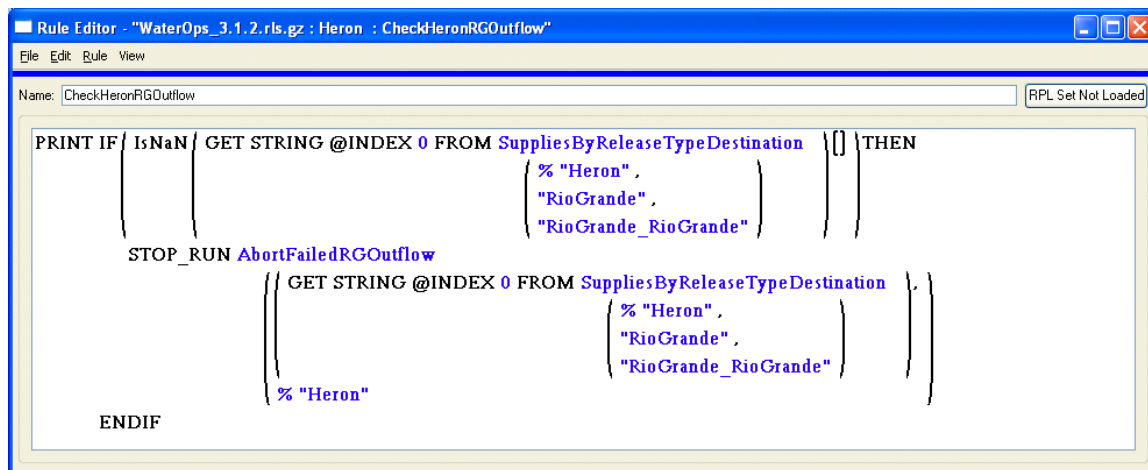


Figure A.42. Rule Policy Language for the CheckHeronRGOutflow Rule

## **A.12. Heron San Juan Accounts**

The rules in the policy group are used to set the accounting supplies for releases of San Juan-Chama Project water from storage accounts at Heron Reservoir. Water is moved to fill the Cochiti rec pool, payback debts, or make deliveries to downstream delivery points or fill downstream allocated storage space. Releases are set based on set priorities until the total outflow has been reached. The total outflow is set with consideration for the different demands, but some accounting supplies may be designated as higher priority to meet needs that are not included as part of the computed total outflow; thus, some accounting supplies may be set differently from what was included for computing the total outflow.

### **A.12.1. SetHeronCochitiRecPoolAccounts**

*Explanation:* This rule sets the accounting supply for filling the Cochiti Rec Pool for the current timestep with consideration for delivery limits for designated periods within the calendar year. The delivery is also restricted to maintain a storage level at Cochiti Lake equal to the input permanent sediment pool. The calculation includes checks against the amount of water in the Cochiti Rec Pool account at Heron Reservoir and the total magnitude of the San Juan release from Heron Dam.

*Rule Execution:* If the value in the HeronCochitiRecPool time series slot in the RuleHasFired data object for the current timestep is a NaN as determined with the user-defined RuleHasNOTFired function in the Priority Functions utility group and the result from the user-defined RuleIsInPriority function in the Priority Functions utility group (Refer to Section A.12.1.1) is true, the rule fires.

*Rule Logic:* The user-defined HeronCochitiRecPoolRelease function in the Heron account Functions utility group is used to compute the Supply for releasing water to the Cochiti Rec Pool. A WITH DO loop is used in the function to create a list of the recreation pool target dates as entered in the CochitiRecPoolMaxVolume periodic slot in the HeronData data object. A second WITH DO loop is used to identify the amount of water released for the Cochiti Rec Pool since the last target date (that is, the value for the CumulativeCochitiRecPoolRelease time series slot in the HeronData data object). The function then computes the release as the amount of water that needs to be released to reach the input maximum volume at the corresponding input target date with consideration for the amount of water released since the last target date. This calculation includes a check against the result from the user-defined MaxCochitiRecPoolVolume function in the Cochiti Account Functions utility group which determines a maximum for the release based on the lesser of the amount of water in the CochitiRecPool account at Heron Reservoir or the available space at Cochiti with consideration for the amount of water in transit to the Cochiti Rec Pool.

The Supply at Heron Reservoir with a release type of CochitiRecPool and a destination of CochitiRecPool\_Cochiti as identified using the GET STRING @INDEX command and the user-defined SuppliesByReleaseTypeDestination function in the Account Lists utility group is set to the result from the previously discussed HeronCochitiRecPoolRelease function or the difference in the San Juan release from Heron Dam and all the supplies for San Juan water released from Heron Dam. This rule includes an additional assignment statement to set the value for the HeronCochitiRecPool time series slot in the RuleHasFired data object for the current timestep to 1.0 using the user-defined RuleHasFiredSlot function in the Priority Functions utility group.

#### **A.12.1.1. RuleIsInPriority**

*Explanation:* This function is used to identify the priority for setting supplies based on input priorities for different release types and different reservoirs in the data object corresponding to the reservoir. The supplies are set up in the model such that there is only one supply for any source and release type combination. So, for a particular source, the supplies can be set based on input priorities for the release type.

*Function Logic:* For Heron Reservoir, a WITH DO loop is used with the user-defined PrioritizedReleaseTypes function in the Priority Functions utility group to create a list of release types and the corresponding priority as input to the ReleaseTypePriority periodic slot in the HeronData data object. The predefined SortPairsAscending function is used to sort the list based on the input priority. A second WITH DO loop is used with the user-defined PrioritizedReservoirs function in the Priority Functions utility group to create a list of the reservoirs (El Vado and Abiquiu) and the corresponding priority as input to the ReservoirPriority periodic slot in the HeronData data object. The predefined SortPairsAscending function is used again to sort the list based on the input priority. A third WITH DO loop is used with the user-defined PrioritizedRuleList function in the Priority Functions utility group to create a list of the rule names. The three following WITH DO loops are used to track whether rules with a higher priority release type or destination reservoir have fired yet. If so, the current rule is in priority.

#### **A.12.2. SetHeronMRGCDPaybacksAbiquiuAccounts**

*Explanation:* This rule sets accounting supplies with a destination of MRGCD\_Abiquiu to payback debts based on input priorities using San Juan water as available.

*Rule Execution:* If the value in the HeronMRGCDPaybacksAbiquiu time series slot in the RuleHasFired data object for the current timestep is a NaN as determined with the user-defined RuleHasNOTFired function in the Priority Functions utility group and the result from the user-defined RuleIsInPriority function in the Priority Functions utility group (Refer to Section A.12.1.1) is true, the rule fires.

*Rule Logic:* The user-defined GetSortedPaybackAccounts function in the Account Lists utility group is used to create a list of supplies and corresponding values to be set. The

values are then set in the rule using a FOREACH loop. Within the function, a WITH DO loop is used with the user-defined PrioritizedAccounts function in the Priority Functions utility group to create a list of accounts with the corresponding priority as input to the AccountReleasePriority periodic slot in the HeronData data object. The predefined SortPairsAscending function is used to sort the list based on the priorities. A second WITH DO loop is used to create a list of the Supplies to be set for all the accounts in that list which have a release type of MRGCDPaybacks and destination of MRGCD\_Abiquiu. An array is then created with the name for each supply and the corresponding payback debt. If water is available to set the Supplies to the payback debt, the supply is set to the minimum of the following: the value for the debt slot for the payback source as determined with the user-defined GetAccountDebt function in the Exchange Functions utility group which references the predefined GetPaybackDebt function, the amount of San Juan water remaining to be allocated as tallied within the function, or the amount of water in storage at Heron Reservoir for the account as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group minus the sum of the supplies from that source as determined with the user-defined SumSupplies function in the Account Summing Function utility group plus the sum of the supplies into that source as determined with the user-defined SumAllSuppliesIntoSource function in the Account Lists utility group.

This rule includes an additional assignment statement to set the value for the HeronMRGCDPaybacksAbiquiu time series slot in the RuleHasFired data object for the current timestep to 1.0 using the user-defined RuleHasFiredSlot function in the Priority Functions utility group.

### **A.12.3. SetHeronMRGCDPaybacksElVadoAccounts**

*Explanation:* This rule sets supplies with a destination of MRGCD\_ElVado and release type of MRGCDPaybacks to pay back debts based on input priorities using San Juan water as available. The code for this rule matches the code for the SetHeronMRGCDPaybacksAbiquiuAccounts Rule except for the destination for the supplies to be set. Refer to the discussion for SetHeronMRGCDPaybacksAbiquiuAccounts Rule for more information on how the assignments are made.

### **A.12.4. SetHeronElVadoDeliveryAccounts**

*Explanation:* This rule sets supplies for all accounts with a destination of a San Juan storage account at El Vado Reservoir as identified with the user-defined MakeDestinationsList function and a release type of ElVadoDelivery to pay back debts based on input priorities using San Juan water as available. The debt is established to move water to ultimately payback letter water debt to Otowi. The code for this rule matches the code for the SetHeronMRGCDPaybacksAbiquiuAccounts Rule except for the destination for the supplies to be set. Refer to the discussion for the

SetHeronMRGCDPaybacksAbiquiuAccounts Rule for more information on how the assignments are made.

#### **A.12.5. SetHeronNMISCAccounts**

*Explanation:* This rule sets supplies with a destination of Jemez\_Jemez and release type of NMISC to pay back debts based on input priorities using San Juan water as available. The code for this rule matches the code for the SetHeronMRGCDPaybacksAbiquiuAccounts Rule except for the release type and destination for the supplies to be set. Refer to the discussion for the SetHeronMRGCDPaybacksAbiquiuAccounts Rule for more information on how the assignments are made. Also, note that for this case of a release type of NMISC and source of Heron Reservoir, the supply is limited based on the amount of water that Albuquerque has in its storage account.

#### **A.12.6. SetHeronOtowiPaybacksAccounts**

*Explanation:* This rule sets supplies with a destination of Otowi\_Otowi and release type of OtowiPaybacks to pay back debts based on input priorities using San Juan water as available. The code for this rule matches the code for the SetHeronMRGCDPaybacksAbiquiuAccounts Rule except for the destination for the supplies to be set. Refer to the discussion for SetHeronMRGCDPaybacksAbiquiuAccounts Rule for more information on how the assignments are made.

#### **A.12.7. SetHeronAlbuquerquePaybacksAccounts**

*Explanation:* This rule sets supplies with a destination of Albuquerque\_Abiquiu and release type of AlbuquerquePaybacks to pay back debts based on input priorities using San Juan water as available. The code for this rule matches the code for the SetHeronMRGCDPaybacksAbiquiuAccounts Rule except for the destination for the supplies to be set. Refer to the discussion for the SetHeronMRGCDPaybacksAbiquiuAccounts Rule for more information on how the assignments are made. Also, note that the supplies set with this rule include consideration for the available account storage at Abiquiu as determined with the user-defined AvailableAccountStorage function with consideration for the loss of San Juan water as referenced with the user-defined SJCLoss function.

### A.12.8. SetHeronWaiverAbiquiuAccounts

*Explanation:* This rule sets supplies for all accounts with a destination of a San Juan storage account at Abiquiu Reservoir as identified with the user-defined MakeDestinationsList to transfer waiver water to the storage accounts through an input number of days.

*Rule Execution:* If the value in the HeronWaiverAbiquiu time series slot in the RuleHasFired data object for the current timestep is a NaN as determined with the user-defined RuleHasNOTFired function in the Priority Functions utility group and the result from the user-defined RuleIsInPriority function in the Priority Functions utility group (Refer to Section A.12.1.1) is true, the rule fires.

*Rule Logic:* The user-defined GetSortedWaiverAccounts function in the Account Lists utility group is used to create a list of supplies and corresponding values to be set. The values are then set in the rule using a FOREACH loop. Within the function, a WITH DO loop is used with the user-defined PrioritizedAccounts function in the Priority Functions utility group to create a list of accounts with the corresponding priority as input to the AccountReleasePriority periodic slot in the HeronData data object. The predefined SortPairsAscending function is used to sort the list based on the priorities. A second WITH DO loop is used to create a list of the Supplies to be set for all those accounts in the previous list which have a destination of a San Juan storage account at Lake Abiquiu as identified in the rule with the user-defined MakeDestinationsList function in the Account Lists utility group. A third WITH DO loop is used to identify the object that corresponds with the destination. An array is then created with the name for each supply and the corresponding supply. The supply is set to the minimum of the following: the WaiverBalance for the account at the previous timestep or the Carry Over if it is January 1 and the start timestep as referenced using the user-defined PreviousWaiverBalance function in the Get Data Functions utility group, the amount of water in storage at Heron Reservoir for the account as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group minus the sum of the supplies from that source as determined with the user-defined SumSupplies function in the Account Summing Function utility group plus the sum of the supplies into that source as determined with the user-defined SumAllSuppliesIntoSource function in the Account Lists utility group, the available account storage space at the destination reservoir as determined with the user-defined AvailableAccountStorage function in the General Account Functions utility group and with consideration for the loss of San Juan water as determined with the user-defined SJCLoss function, or the amount of San Juan water remaining to be allocated as tallied within the function. The available water is computed in the rule as the result from user-defined ReconcileSJOutflow function in the General Account Functions utility group and the sum of all supplies for San Juan water at Heron Reservoir and is restricted based on the result for the user-defined HeronWaiverRelease function in the Heron Account Functions utility group.

The HeronWaiverRelease function identifies the total waiver release for the rest of the year. If the current timestep is before the waiver date determined with the user-defined WaiverDate function in the Get Data Functions utility group and the value for the WaiverSwitch time series slot in the HeronData data object is equal to 1.0 as determined with the user-defined PreviousYearIsWaiverYear function in the General Boolean Functions utility group and the total waiver balance as determined with the user-defined TotalWaiverBalance function in the Account Summing Functions utility group is greater than zero, the release is computed as the average outflow required to release the total waiver balance before the waiver date. Otherwise, the result for the HeronWaiverRelease function is set 0.0 cfs.

The rule includes an additional assignment statement to set the value for the HeronWaiverAbiquiu time series slot in the RuleHasFired data object for the current timestep to 1.0 with the user-defined RuleHasFiredSlot function in the Priority Functions utility group.

#### **A.12.9. SetHeronWaiverElVadoAccounts**

*Explanation:* This rule sets supplies for all accounts with a destination of a San Juan storage account at El Vado Reservoir to transfer waiver water to the storage accounts for an input number of days. The code for this rule matches the code for the SetHeronWaiverAbiquiuAccounts Rule except for the destination for the supplies to be set. Refer to the discussion for the SetHeronWaiverAbiquiuAccounts Rule for more information on how the assignments are made.

#### **A.12.10. SetHeronAccountFillAbiquiuAccounts**

*Explanation:* This rule sets supplies for all accounts with a destination of a San Juan storage account at Abiquiu Reservoir as identified with the user-defined MakeDestinationsList to fill the downstream storage accounts with San Juan water as available. The calculation includes consideration for input maximum fill amounts and corresponding target dates.

*Rule Execution:* If the value in the HeronAccountFillAbiquiu time series slot in the RuleHasFired data object for the current timestep is a NaN as determined with the user-defined RuleHasNOTFired function in the Priority Functions utility group and the result from the user-defined RuleIsInPriority function in the Priority Functions utility group (Refer to Section A.12.1.1) is true, the rule fires.

*Rule Logic:* The user-defined GetSortedStorageAccounts function in the Account Lists utility group is used to create a list of supplies and corresponding values to be set. The values are then set in the rule using a FOREACH loop. Within the function, a WITH DO loop is used with the user-defined PrioritizedAccounts function in the Priority Functions utility group to create a list of accounts with the corresponding priority as input to the



AccountReleasePriority periodic slot in the HeronData data object. The predefined SortPairsAscending function is used to sort the list based on the priorities. A second WITH DO loop is used to create a list of the Supplies to be set for all those accounts in the previous list which have a destination of a San Juan storage account at Lake Abiquiu as identified in the rule with the user-defined MakeDestinationsList function in the Account Lists utility group. A third WITH DO loop is used to identify the object that corresponds with the destination. An array is then created with the name for each supply and the corresponding value for the supply. If water is available, the supply is set to the minimum of the following: the amount of water in storage at Heron Reservoir for the account as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group minus the sum of the supplies from that source as determined with the user-defined SumSupplies function in the Account Summing Function utility group plus the sum of the supplies into that source as determined with the user-defined SumAllSuppliesIntoSource function in the Account Lists utility group, the available account storage space at the destination reservoir as determined with the user-defined AvailableAccountStorage function in the General Account Functions utility group and with consideration for the loss of San Juan water as determined with the user-defined SJCLoss function, or the amount of San Juan water remaining to be allocated as tallied within the function. The available water is computed in the rule as the result from user-defined ReconcileSJOutflow function in the General Account Functions utility group and the sum of all supplies for San Juan water at Heron Reservoir and is restricted based on the result for the user-defined HeronAccountFillRelease function in the Heron Account Functions utility group.

The HeronAccountFillRelease function computes an average release to either fill downstream storage accounts or provide the maximum fill volume at the next target date as entered in the AccountFillMaxVolume periodic slot in the HeronData data object. The maximum fill volume is based on the value input to the AccountFillMaxVolume periodic slot and includes consideration for the cumulative release to date since the last target date as input to the CumulativeAccountFillRelease time series slot in the HeronData data object for the previous timestep (Refer to the SetCumulativeAccountFillRelease Rule). Several separate functions are used within the user-defined HeronAvailableDownstreamAccountStorage function in the Heron Account Functions utility group to compute the requirement to fill downstream storage accounts. The available space for the different storage accounts is computed with consideration for supplies into and from the source for the current timestep and with consideration for payback debts.

The rule includes an additional assignment statement to set the value for the HeronAccountFillAbiquiu time series slot in the RuleHasFired data object for the current timestep to 1.0 using the user-defined RuleHasFiredSlot function in the Priority Functions utility group.

### **A.12.11. SetHeronAccountFillElVadoAccounts**

*Explanation:* This rule sets supplies for all accounts with a destination of a San Juan storage account at El Vado Reservoir to fill the downstream storage accounts. The code for this rule matches the code for the SetHeronAccountFillAbiquiuAccounts Rule except for the destination for the supplies to be set. Refer to the discussion for the SetHeronAccountFillAbiquiuAccounts Rule for more information on how the assignments are made.

#### **A.12.12. SetHeronAccountDeliveryAbiquiuAccounts**

*Explanation:* This rule sets supplies for all accounts with a destination of a San Juan storage account at Abiquiu Reservoir to deliver water to the downstream storage accounts. The code for this rule matches the code for the SetHeronAccountFillAbiquiuAccounts Rule except for the release type which is AccountDelivery as opposed to AccountFill for the supplies to be set. Refer to the discussion for the SetHeronAccountFillAbiquiuAccounts Rule for more information on how the assignments are made.

#### **A.12.13. SetHeronAccountDeliveryElVadoAccounts**

*Explanation:* This rule sets supplies for all accounts with a destination of a San Juan storage account at El Vado Reservoir to deliver water to downstream storage accounts. The code for this rule matches the code for the SetHeronAccountFillAbiquiuAccounts Rule except the destination is storage accounts at El Vado Reservoir and the release type is AccountDelivery as opposed to AccountFill for the supplies to be set. Refer to the discussion for the SetHeronAccountFillAbiquiuAccounts Rule for more information on how the assignments are made.

#### **A.12.14. SetHeronReleaseToMRGCDAccounts**

*Explanation:* This rule sets the supplies for the accounts to release San Juan water to MRGCD's storage account at El Vado Reservoir as available.

*Rule Execution:* If the value in the HeronReleaseToMRGCD time series slot in the RuleHasFired data object for the current timestep is a NaN as determined with the user-defined RuleHasNOTFired function in the Priority Functions utility group and the result from the user-defined RuleIsInPriority function in the Priority Functions utility group (Refer to Section A.12.1.1) is true, the rule fires.

*Rule Logic:* The user-defined GetSortedReleaseToMRGCDAccounts function in the Account Lists utility group is used to create a list of supplies and corresponding values to be set. The values are then set in the rule using a FOREACH loop. Within the function, a WITH DO loop is used with the user-defined PrioritizedAccounts function in the Priority Functions utility group to create a list of accounts with the corresponding priority

as input to the AccountReleasePriority periodic slot in the HeronData data object. The predefined SortPairsAscending function is used to sort the list based on the priorities. A second WITH DO loop is used to create a list of the Supplies to be set for the accounts in the previous list which have a release type of ReleaseToMRGCD and a destination of MRGCD\_ElVado. An array is then created with the name for each supply and the corresponding value for the supply. If water is available, the supply is set to the minimum of the following: the amount of water in storage at Heron Reservoir for the account as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group minus the sum of the supplies from that source as determined with the user-defined SumSupplies function in the Account Summing Function utility group plus the sum of the supplies into that source as determined with the user-defined SumAllSuppliesIntoSource function in the Account Lists utility group, the available account storage space at the destination reservoir as determined with the user-defined AvailableAccountStorage function in the General Account Functions utility group and with consideration for the loss of San Juan water as determined with the user-defined SJCLoss function, or the amount of San Juan water remaining to be allocated as tallied within the function. The available water is computed in the rule as the result from user-defined ReconcileSJOutflow function in the General Account Functions utility group minus the sum of all supplies for San Juan water at Heron Reservoir.

The rule includes an additional assignment statement to set the value for the HeronReleaseToMRGCD time series slot in the RuleHasFired data object for the current timestep to 1.0 using the user-defined RuleHasFiredSlot function in the Priority Functions utility group.

### ***A.13. Preparatory ElVado***

The rules in this policy group are used to compute a preliminary value for a total release of San Juan-Chama Project water from El Vado Dam, adjust the waiver balance for accounts that have storage space at El Vado or Abiquiu Reservoir, and reset the waiver balance to 0.0 acre-ft after the waiver period is over and transfer the waiver balance to the federal pool at Heron Reservoir.

#### **A.13.1. ComputeElVadoSJRelease**

*Explanation:* This rule computes a preliminary value for the release of San Juan water from El Vado Dam. The computation includes consideration for pass-through flows, flows to payback debts downstream, and flows to fill allocated downstream storage space. The calculation also includes the release of water from Reclamation's account to meet downstream target flows and the release of water from MRGCD's account to meet downstream demands. The result is checked against a minimum release.

*Rule Execution:* If the value for the SJOutflow time series slot in the ElVadoData data object for the current timestep is a NaN, the rule fires.

*Rule Logic:* The user-defined TotalElVadoSJRelease function in the ElVado Account Functions utility group is used to compute the release of San Juan water. This function sums the result from the following separate user-defined functions in the ElVado Account Functions utility group and then assures that the summation does not exceed a determined maximum. The TotalElVadoFlowThruAccounts function sums the flow through accounts at El Vado. The ElVadoOtowiDebtRelease function sums the supplies at El Vado with a release type of OtowiPaybacks. Unless it is the rafting release season, the ElVadoSJStorageAccountsRelease function references the user-defined PreliminaryElVadoSJStorageAccountsRelease function to sum the supplies with a release type of AlbuquerquePaybacks and destination of Albuquerque\_Abiquiu. If it is the rafting release season, the function is 0.0 cfs. The ElVadoMRGCDDebtRelease function computes the release for Reclamation to payback debts to MRGCD. The ElVadoReclamationRelease function computes the release of water in Reclamation's storage account at El Vado Reservoir with consideration for the available space in Reclamation's storage account at Abiquiu Reservoir. The ComputeElVadoMRGCDSJRelease function computes the release of MRGCD's San Juan-Chama Project water needed to meet the downstream MRGCD demand. The user-defined MaximumSJOutflowElVado function in the Get Date Functions utility group is used to determine the maximum San Juan release.

A WITH DO loop is used in the rule to set the value for the SJOutflow time series slot in the ElVadoData data object to the result from the user-defined TotalElVadoSJRelease function unless that result is less than the minimum outflow from Heron Dam as determined with the user-defined MinOutflow function in the Get Data Functions utility group minus the value in the RGOutflow time series slot in the ElVadoData data object as referenced using the user-defined RGOutflow function in the General Account Functions utility group. In that case, if there is also sufficient space available in the storage accounts at Abiquiu with consideration for the Otowi debt, the release is set higher to the difference in the minimum outflow and the Rio Grande outflow. An IF THEN ELSE statement is also included to set the release to 0.0 cfs if the result from the user-defined TotalElVadoSJRelease function is less than 1.0 cfs. Refer to Figures A.43 for a diagram that shows the different aspects of the computed initial release of San Juan-Chama Project water from El Vado Dam.

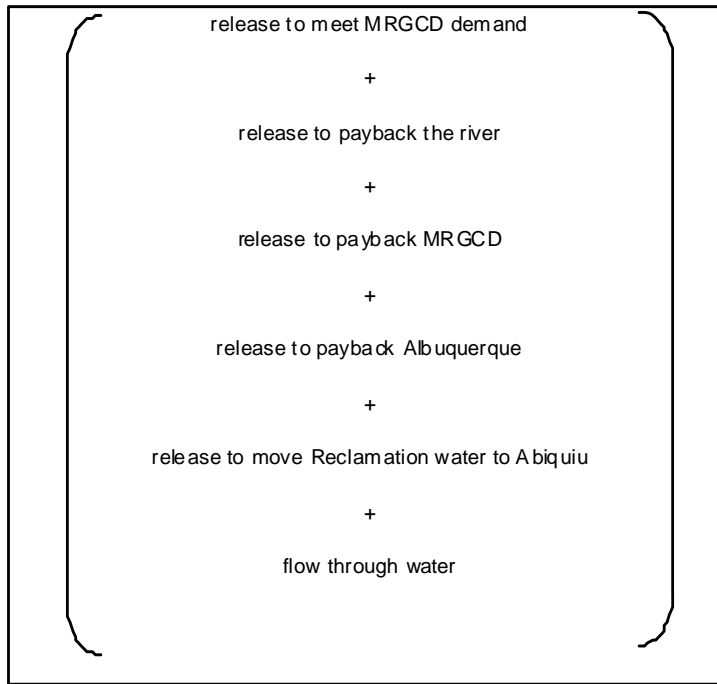


Figure A.43. Components of Initial Total Release of SJC Water from El Vado Reservoir

### A.13.2. SetWaiverBalance

*Explanation:* This rule updates the waiver balance, for the accounts that have allocated storage space at El Vado or Abiquiu Reservoirs, by adjusting the previous waiver balance for any release of waiver water from Heron Dam at the current timestep. On January 1<sup>st</sup>, the previous waiver balance is equal to the previous account storage (on December 31<sup>st</sup>).

*Rule Execution:* If the AlbuquerqueWaiverBalance time series slot in the HeronData data object for the current timestep is a NaN, the rule fires.

*Rule Logic:* If the result from the user-defined WaiversAreInEffect function in the General Boolean functions utility group is TRUE, the values for the WaiverBalance, for contractors that have allocated storage space at El Vado or Abiquiu Reservoirs, are set using a FOREACH loop. The WaiversAreInEffect function checks to see if the value for the WaiverSwitch in the HeronData data object referenced with the user-defined PreviousYearWaiverSwitch function in the Get Data Functions utility group is equal to 1.0 as determined with the user-defined PreviousYearIsWaiverYear function in the General Boolean Functions utility group and also whether the current timestep is before December 31 plus the value in the WaiverDate table slot in the HeronData data object, or that value plus 1.0 if the result from the predefined LeapYear function is TRUE, referenced using the user-defined WaiverDay function in the Get Data Functions utility group and computed using the user-defined WaiverDate function in the Get Data Functions utility group.

If the current timestep is January 1, the WaiverBalance is set to the amount of water in storage at Heron Reservoir for the account as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group minus the sum of the supplies at Heron Reservoir for that account with a release type of Waiver for the current timestep. Otherwise (i.e. it is not January 1), if the account is NOT Albuquerque, the WaiverBalance is set to the WaiverBalance for the account for the previous timestep as referenced using the user-defined PreviousWaiverBalance function in the Get Data Functions utility group minus the sum of the supplies at Heron Reservoir for that account with a release type of Waiver for the current timestep. If the account is Albuquerque, the calculation is the same except the sum of the supplies at Heron Reservoir for that account with a release type of MRGCDPaybacks and destination of MRGCD\_ElVado for the current timestep is also subtracted. If none of those criteria applies, the value is set to 0.0 cfs.

### **A.13.3. ZeroWaivers**

*Explanation:* This rule is used to set accounting supplies to transfer the waiver balance for contractors back to the federal pool after the waiver period is over, for the accounts that have storage space at El Vado or Abiquiu Reservoirs, and subsequently set the separate tracked waiver balance for those accounts for the current timestep to 0.0 acre-ft.

*Rule Execution:* The rule fires if the current timestep is equal to the date in the EndOfWaiverPeriod column in the WaiverDate table slot in the HeronData data object, plus 1 day, where the waiver date is referenced with the user-defined WaiverDate function in the Get Data Functions utility group and determined with the predefined GetTableDate function AND the value for the WaiverSwitch in the HeronData data object for the previous year referenced using the user-defined PreviousYearWaiverSwitch function in the Get Data Functions utility group is equal to 1.0 as determined with the user-defined PreviousYearIsWaiverYear function in the General Boolean Functions utility group.

*Rule Logic:* The first FOREACH loop is used to set the accounting supplies, for accounts that have allocated storage space at El Vado or Abiquiu Reservoirs, to transfer water from the accounts at Lake Heron to the federal pool equal to the WaiverBalances for the accounts for the previous timestep as referenced using the user-defined PreviousWaiverBalance function in the Get Data Functions utility group. The second FOREACH loop is used to set the WaiverBalances for those same accounts for the current timestep to 0.0 acre-ft.

### **A.14. ElVado**

The rules in this policy group are used to ultimately determine the outflow from El Vado Dam with consideration for any release to maintain storage below the maximum pool

elevation and maintain releases below downstream channel capacities. The accounting supply for the release of Rio Grande water from El Vado Dam is set.

### **A.14.1. ElVadoOutflow**

*Explanation:* Unless a total outflow for El Vado Dam has been input for the current timestep, the value is set to the sum of the predetermined Rio Grande release and the San Juan release, or the minimum release if it is larger. That total release is checked to see if it is physically legitimate based on the outlet works, and if not, the release is reset to reflect the restriction for the outlet works.

*Rule Execution:* If the value for the Outflow time series slot in the ElVado storage reservoir object for the current timestep is a NaN, the rule fires.

*Rule Logic:* This rule contains an IF THEN ELSE statement to see if the value in the TotalOutflow time series slot in the ElVadoData data object for the current timestep as referenced with the user-defined TotalOutflow function in the Get Data Functions utility group is a NaN. If not, the value for the Outflow time series slot in the ElVado storage reservoir object for the current timestep is set to that input TotalOutflow value. Otherwise, the Outflow is set using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1) with the sum of the RGOutflow (Refer to the ComputeElVadoRGRelease Rule) and SJOutflow (Refer to the ComputeElVadoSJRelease Rule) for El Vado Dam as the predetermined outflow, or the minimum outflow if it is larger as determined using the user-defined MinOutflow function in the Get Data Functions utility group.

### **A.14.2. ElVadoRafting**

*Explanation:* This rule is not used and would need to be reviewed before it is used but could maybe be used to increase the determined total outflow from El Vado Dam to include a computed rafting release.

*Rule Execution:* If this rule has not fired as checked against the predefined HasRuleFiredSuccessfully function, the rule fires.

*Rule Logic:* The value for the Outflow time series slot in the ElVado storage reservoir object for the current timestep is set to the currently set value for the Outflow time series slot for the current timestep plus the result from the user-defined ElVadoRaftingRelease function in the Rafting Release Functions utility group. Refer to the discussion for the SetRaftingRelease Rule for more information on this function.

### A.14.3. ElVadoIndianPuebloWaterRight

*Explanation:* If it is the spring runoff season or the summer irrigation season, the value for the Indian Met is recorded. If the storage in the Rio Grande storage account is less than zero, the Indian Met is set to zero. If the Indian Call is greater than zero and the storage in the IndianStorage account at El Vado Reservoir is less than or equal to the Indian storage requirement, the value is then set to the minimum of the Indian Call, the Indian storage requirement, or the amount of water in the Indian storage account at El Vado Reservoir.

*Rule Execution:* If this rule has not fired as checked against the predefined HasRuleFiredSuccessfully function, the rule fires.

*Rule Logic:* If the result from either the user-defined ElVadoSpringRunoff or the ElVadoSummerIrrigationSeason functions in the ElVado Account Functions utility group is true, the value for the Indian Met time series slot in the Indian data object for the current timestep is set. The ElVadoSpringRunoff function checks to see if the current timestep is between March 15 and the target fill date (inclusive) (Refer to Section A.9.7.1 for more information on the logic for the TargetFillDate function used to determine the target fill date for El Vado Reservoir). The ElVadoSummerIrrigationSeason function checks to see if it is the El Vado rafting season or if it is after the target fill date and before the end of October (Refer to the discussion for the SetRaftingRelease Rule for more information on the logic in the ElVadoRaftingSeason function).

An interior IF THEN ELSE statement is used to see if the storage in the Rio Grande storage account at El Vado Reservoir as referenced using the user-defined PreviousAccountStorage function in the General Account Functions utility group is less than or equal to 0.0 acre-ft. If so, the Indian Met value is set to 0.0 cfs. Otherwise, if the storage in the IndianStorage account at El Vado Reservoir as referenced with the user-defined PreviousAccountStorage function is less than or equal to the result from the user-defined IndianStorageRequirement function (Refer to the discussion for the ElVadoIndianAccountTransfer Rule for more information on this function) and the value for the Indian Call time series slot in the Indian data object for the current timestep (Refer to Indian Storage Requirement Release Rule) as referenced using the user-defined IndianCall function in the Get Data Functions utility group is greater than 0.0 cfs, the Indian Met value is then set to the minimum of the Indian Call, the Indian Storage Requirement converted to a flowrate using the predefined VolumeToFlow function, or the Storage in the IndianStorage account at El Vado Reservoir at the previous timestep converted to a flowrate.



#### **A.14.4. ElVadoFloodControl**

*Explanation:* If the pool elevation at El Vado Reservoir is higher than the input maximum, the outflow is set to the release required to reduce the pool elevation to that maximum pool elevation or the maximum release from the outlet works if that is lower. A check is also included to assure the minimum release is exceeded.

*Rule Execution:* The rule fires if the value for the Pool Elevation time series slot in the ElVado storage reservoir object for the current timestep is greater than the value in the MaxAllowableElevation periodic slot in the ElVadoData data object for the previous timestep and if this rule has not already fired for the current timestep as checked against the predefined HasRuleFiredSuccessfully function.

*Rule Logic:* The value for the Outflow time series slot in the ElVado storage reservoir object for the current timestep is set to the minimum of the result from the user-defined ComputeMaxOutflow function in the General Compute Functions utility group which references the predefined GetMaxOutflowGivenInflow function or the outflow required to reduce the pool elevation to the value in the MaxAllowableElevation periodic slot in the ElVadoData data object for the previous timestep as computed using the predefined SolveOutflow function. That result is checked to assure it exceeds the minimum outflow from El Vado Dam as determined with the user-defined MinOutflow function in the Get Data Functions utility group.

#### **A.14.5. ElVadoChannelCapacity**

*Explanation:* This rule adjusts the outflow from El Vado Dam to comply with the input downstream channel capacity if the predetermined outflow exceeds that capacity. The values for the Indian Met and Texas Met slots are then adjusted accordingly if necessary. Note that this rule fires after the ElVadoFloodControl rule, so the reservoir level will rise above the maximum pool elevation if required to keep downstream flows less than the channel capacity.

*Rule Execution:* The rule fires if the currently set value in the Outflow time series slot in the ElVado data object for the current timestep is greater than the maxRelease in the ReservoirData table slot in the ElVadoData data object as referenced using the user-defined MaxOutflow function in the Get Data Functions utility group and if this rule has not already fired for the current timestep as checked with a reference against the predefined HasRuleFiredSuccessfully function.

*Rule Logic:* This rule includes four assignment statements. The value for the Indian Met time series slot in the Indian data object for the current timestep is set with the user-defined ElVadoIndianMet function in the ElVado Compact Functions utility group. This function includes an IF THEN ELSE statement. If the value for the Indian Call time

series slot in the Indian data object is greater than the maxRelease in the ReservoirData table slot in the ElVadoData data object as referenced with the user-defined MaxOutflow function, the value for the Indian Met is set to that maxRelease. Otherwise, it is set to the value for the Indian Call for the current timestep.

The value for the Texas Met time series slot in the RioGrandeCompact data object for the current timestep is set using the user-defined ElVadoTexasMet function in the ElVado Compact Function utility group. This function includes an exterior IF THEN ELSE statement to see if the value for the Indian Call time series slot for the current timestep plus the value in the Texas Call time series slot in the RioGrandeCompact data object for the current timestep is greater than the maxRelease. If not, the value for the Texas Met slot is set to the value in the Texas Call time series slot for the current timestep. If so, an interior IF THEN ELSE statement checks to see if just the value for the Indian Call slot is greater than the maxRelease. If so, the Texas Met slot is set to 0.0 cfs. If not, the value is set to the maxRelease minus the value in the Indian Call slot. For the third assignment statement, the value for the Outflow time series slot in the ElVado storage reservoir object is set to the maxRelease.

#### **A.14.6. SetRioGrandeAccountCompactVII**

*Explanation:* This rule sets the accounting supply for the release of Rio Grande water from El Vado Dam for the case when Article VII of the Compact is in effect (i.e. all inflows are bypassed as not needed for P&P or Emergency Drought storage). The value is essentially set to the Rio Grande inflow minus the computed inflow to Emergency Drought storage and minus the inflow to P&P storage. The result is checked against the reconciled Rio Grande outflow and the minimum Rio Grande release.

*Rule Execution:* The rule fires if the value for the RioGrandeElVadoToRioGrandeBlwElVado supply for the current timestep is a NaN and if the stipulations of Article VII of the Compact are in effect as indicated by a value of 1.0 for the CompactVIISwitch time series slot in the RioGrandeCompact data object for the current timestep as determined with the user-defined CompactVIIinEffect function in the ElVado Compact Functions utility group.

*Rule Logic:* If the Outflow slot for the El Vado storage reservoir object is equal to the result from the user-defined ComputeElVadoFloodControl function and the ElVadoFloodControl rule has fired successfully for the current timestep as identified using the predefined HasRuleFiredSuccessfully function, the accounting supply for the release of Rio Grande water from El Vado Dam for the current timestep is set to the result from the ReconcileRGOutflow function for the El Vado object.

Otherwise, the value for the supply is set to the minimum of the result from the user-defined ReconcileRGOutflow function in the General Account Functions utility group or the result from the user-defined CurrentRGInflow function in the General Compute Functions utility group minus the result from the MRGCDDroughtInflow function in the

ElVado Account Functions utility group minus the result from the SupplementalESAINflow function in the ElVado Account Functions utility group minus the result from the IndianStorageInflow function in the ElVado IndianStorage Account Functions utility group plus the result from the IndianStorageTransferRelease function in the ElVado Account Functions utility group plus the minimum of the result from the ElVadoMRGCDDReleaseRG function or the amount of water in storage at El Vado Reservoir for the Rio Grande account as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group plus the result from the user-defined PreviousAccountGainLoss function in the Get Data Functions utility group plus the amount of water in storage at El Vado Reservoir for the Rio Grande account as determined with the user-defined PreviousAccountStorage function if that previous account storage is less than zero. The minimum from the previous two values is then checked to assure it exceeds the lesser of the minimum Rio Grande outflow as determined with the user-defined MinRGOutflow function in the Get Data Functions utility group or the result from the ReconcileRGOutflow function.

The equation in the ReconcileRGOutflow function for El Vado Reservoir includes an IF THEN ELSE statement to see if the value for the RGOutflow time series slot in the ElVadoData data object for the current timestep as referenced with the user-defined RGOutflow function in the General Account Functions utility group is greater than the value in the Outflow time series slot in the ElVado storage reservoir object. If so, the function result is set to that Outflow for the ElVado storage reservoir object. Otherwise, a second IF THEN ELSE statement is used to see if it is the irrigation season (i.e. the current timestep is between March 1 and October 31 (inclusive)) as determined using the user-defined IrrigationSeason function in the Date Functions utility group and if the value for the Outflow time series slot for the ElVado storage reservoir object is greater than the sum of the values in the RGOutflow and SJOutflow time series slots in the ElVadoData data object and if the value in the RGOutflow time series slot for the current timestep is less than the value in the MRGCDDemand time series slot in the ElVadoData data object for the current timestep and if the amount of water in Storage at El Vado Reservoir for the MRGCD account as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group plus the sum of the supplies into that account as determined with the user-defined SumAllSuppliesIntoSource function in the Account Lists utility group plus the value in the RGOutflow time series slot is all greater than or equal to the value in the MRGCDDemand time series slot. If all those criteria are not satisfied, the function result is set to the value for the Outflow time series slot for the ElVado storage reservoir object minus the SJOutflow time series slot for the current timestep. If so, the function result is set to the outflow from the ElVado storage reservoir object minus the quantity of the value of the SJOutflow time series slot minus the quantity of the Outflow from the ElVado storage reservoir object minus the quantity of the value for the RGOutflow time series slot for the current timestep plus the value of the SJOutflow time series slot all for the current timestep. If Article VII of the Compact is NOT in effect, the calculation is the same except the final result is checked to see if it exceeds the minimum Rio Grande outflow as referenced using the user-defined MinRGOutflow function in the Get Data Functions utility group, and the result is set to

the RGOutflow if the value in the Outflow time series slot for the ElVado storage reservoir object is greater than the sum of the RGOutflow and SJOutflow.

The CurrentRGInflow function computes the inflow for El Vado Reservoir as the value for the RGOutflow time series slot in the HeronData data object for the current timestep plus the value for the Local Inflow time series slot in the ElVadoLocalInflow reach object for the current timestep plus the value for the Seepage time series slot in the Heron storage reservoir object for the previous timestep if it is not a NaN.

#### **A.14.7. SetRemainingElVadoRGAccountsCompactVII**

*Explanation:* If the stipulations of Article VII of the Compact are in effect, this rule sets the accounting supplies for releases from the other Rio Grande accounts at El Vado Reservoir (MRGCDDrought, Indian Storage, and SupplementalESA).

*Rule Execution:* The rule fires if this rule has not already fired for the current timestep as checked with a reference against the predefined HasRuleFiredSuccessfully function and if the stipulations of Article VII of the Compact are in effect as indicated by a value of 1.0 for the CompactVIISwitch time series slot in the RioGrandeCompact data object for the current timestep as determined using the user-defined CompactVIIinEffect function in the ElVado Compact Functions utility group.

*Rule Logic:* The user-defined ComputeElVadoRGSuppliesListCompactVII function in the Account Lists utility group is used to create a list of supplies and corresponding values to be set. The values are then set in the rule using a FOREACH loop. Within the function, a WITH DO loop is used with the user-defined SuppliesByReleaseTypeDestination function in the Account Lists utility group to create a list of supplies for accounts at El Vado Reservoir with a release type of RioGrande. A second WITH DO loop is used to edit the list to delete the supply for the RioGrande account (already set with the SetRioGrandeAccountCompactVII Rule). An array is then created with the name for each supply and the corresponding value for the supply computed using the user-defined ComputeElVadoRGSuppliesCompactVII function in the ElVado Account Functions utility group. That function includes separate equations for computing the supplies for releases from the MRGCDDrought, IndianStorage, and SupplementalESA accounts.

#### **A.14.8. SetElVadoRGAccounts**

*Explanation:* If the stipulations of Article VII of the Compact are NOT in effect, this rule sets the accounting supplies for the release of Rio Grande water from El Vado Reservoir with four separate supplies included for releases from the MRGCDDrought, Indian Storage, SupplementalESA, and Rio Grande accounts at El Vado Reservoir. This rule serves as the alternative to the SetRioGrandeAccountCompactVII and

SetRemainingElVadoRGAccountsCompactVII rules which set the supplies if Article VII is in effect.

*Rule Execution:* The rule fires if the value for the Supply for the account at El Vado Reservoir with a release type of Rio Grande and destination of RioGrande\_RioGrande referenced using the user-defined SuppliesByReleaseTypeDestination function in the Account Lists utility group for the current timestep is a NaN and if the stipulations of Article VII of the Compact are NOT in effect as indicated by a value NOT equal to 1.0 in the CompactVIISwitch time series slot in the RioGrandeCompact data object for the current timestep as determined with the user-defined CompactVIIinEffect function in the ElVado Compact Functions utility group.

*Rule Logic:* The user-defined ComputeElVadoRGSuppliesList function in the Account Lists utility group is used to create a list of supplies and corresponding values to be set. The values are then set in the rule using a FOREACH loop. Within the function, a WITH DO loop is used with the user-defined SuppliesByReleaseTypeDestination function to create a list of supplies for accounts at El Vado Reservoir with a release type of RioGrande. An array is created with the name for each supply and the corresponding value for the supply computed using the user-defined ComputeElVadoRGSupplies function in the ElVado Account Functions utility group. That function includes four separate equations for computing the supplies for the SupplementalESA, MRGCDDrought, IndianStorage, and RioGrande accounts at El Vado Reservoir.

#### **A.14.9. CheckElVadoRGOutflow**

*Explanation:* Aborts the simulation if the supply for the release of Rio Grande water from the account at El Vado Reservoir was not set. This rule helps with debugging because a simulation will stop if the Rio Grande supply was not set and the problem and any missing input can be identified by backtracking in the calculations from the point in the simulation that the run aborted.

*Rule Execution:* There are no execution constraints for this rule.

*Rule Logic:* If the value for the Supply at El Vado Reservoir with a release type of Rio Grande and destination of RioGrande\_RioGrande is a NaN, the STOP\_RUN command is used to stop the simulation, and the PRINT statement is used to print a comment to the Diagnostics Output window that the run stopped because the account slot value was not set.

#### **A.15. ElVado San Juan Accounts**

The rules in this policy group are used to set the accounting supplies for releases of San Juan-Chama Project water, as available, from storage at El Vado Reservoir to payback debts or fill allocated downstream storage space. Releases are set based on set priorities

until the total outflow has been reached. The total outflow is set with consideration for the different demands.

### **A.15.1. SetElVadoFlowThroughAccounts**

*Explanation:* This rule sets accounting supplies for all the flowthrough accounts at El Vado Reservoir.

*Rule Execution:* If the value in the ElVadoFlowThrough time series slot in the RuleHasFired data object for the current timestep is a NaN as determined with the user-defined RuleHasNOTFired function in the Priority Functions utility group and the result from the user-defined RuleIsInPriority function in the Priority Functions utility group (Refer to Section A.15.1.1) is true, the rule fires.

*Rule Logic:* The user-defined GetSortedFlowThruAccounts function in the Account Lists utility group is used to create a list of supplies and corresponding values to be set. The values are then set in the rule using a FOREACH loop. Within the function, a WITH DO loop is used with the predefined AccountNamesFromObjReleaseDestination function to create a list of accounts at El Vado Reservoir with a release type of FlowThrough. A second WITH DO loop is used to create a list of the supplies to be set for all those accounts which have a release type of FlowThrough. An array is then created with the name for each supply and the corresponding supply value. If water is available, values for the supplies are set. A third WITH DO loop is used to identify an upstream slot name to be referenced in the upstream object, input as Heron Seepage in the rule, and a fourth WITH DO loop is used to identify the upstream reservoir.

The value for each supply is set as follows: first, a list of supplies is created from the previously generated list of accounts using the user-defined AllSuppliesIntoBySourcesReleaseTypeDestination function in the Account Lists utility group with the designated source as the upstream object. The values for those supplies at the timestep equal to the current timestep minus the number of timesteps input to the Lags table slot in the SanJuanChamaRules data object as referenced using the user-defined SJCLags function in the Get Data Functions utility group are summed utilizing the user-defined SumSuppliesAtDate function in the Account Summing Functions utility group. The loss of San Juan water between the upstream reservoir and El Vado Reservoir as input to the Losses table slot in the SanJuanChamaRules data object is applied to the previous summation. Finally, the water in storage at El Vado Reservoir for the account at the previous timestep as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group is added. The predefined Min function is used to restrict that final result to the amount of San Juan water remaining to be allocated as tallied within the function. The available water is computed in the rule as the result from user-defined ReconcileSJOutflow function in the General Account Functions utility group minus the sum of all supplies for San Juan water at El Vado Reservoir.

The rule includes an additional assignment statement to set the value for the ElVadoFlowThrough time series slot in the RuleHasFired data object for the current timestep to 1.0 using the user-defined RuleHasFiredSlot function in the Priority Functions utility group.

#### **A.15.1.1. RuleIsInPriority**

*Explanation:* This function is used to identify the priority for setting supplies based on input priorities for different release types in the data object for the reservoir. The supplies are set up in the model such that there is only one supply for any source and release type combination. So, for a particular source, the supplies can be set based on input priorities for the release type.

*Function Logic:* For reservoirs other than Heron Reservoir, a WITH DO loop is used with the user-defined PrioritizedReleaseTypes function in the Priority Functions utility group to create a list of release types and the corresponding priority as input to the ReleaseTypePriority periodic slot in the ElVadoData data object. The predefined SortPairsAscending function is used to sort the list based on the input priority. The two following WITH DO loops are used to track whether rules with a higher priority release type have fired yet. If so, the current rule is in priority.

#### **A.15.2. SetElVadoMRGCDAccount**

*Explanation:* This rule sets the value for the accounting supply for the release of MRGCD San Juan-Chama Project water from El Vado to the MRGCD storage account at Abiquiu Reservoir (The MRGCD account at Abiquiu actually serves as an operating pool for MRGCD's delivery of water to meet their demand in the Middle Valley).

*Rule Execution:* The rule fires if the value in the ElVadoMRGCD time series slot in the RuleHasFired data object for the current timestep is a NaN as determined with the user-defined RuleHasNOTFired function in the Priority Functions utility group and the result from the user-defined RuleIsInPriority function in the Priority Functions utility group (Refer to Section A.15.1.1) is true.

*Rule Logic:* The value for the supply is set to the minimum of the following three computed values: 1) the value for the ComputedMinMRGCDRelease time series slot in the ElVadoData data object minus the values for the two supplies at El Vado Reservoir with a release type of Rio Grande and destinations of RioGrande\_RioGrande and RioGrande\_MRGCD, 2) The water in storage at El Vado Reservoir for the MRGCD account at the previous timestep as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group plus the sum of all the supplies at El Vado Reservoir and Heron Reservoir with a destination of MRGCD\_ElVado each summed utilizing the user-defined SumSuppliesAtDate function in the Account Summing Functions utility group plus the Gain Loss for the MRGCD account at El Vado Reservoir for the current timestep minus the sum of all the supplies at

El Vado Reservoir with MRGCD as a source, and 3) The amount of San Juan water available to be allocated computed as the result from user-defined ReconcileSJOutflow function in the General Account Functions utility group minus the sum of all the supplies for San Juan water at El Vado Reservoir.

The rule includes an additional assignment statement to set the value for the ElVadoMRGCD time series slot in the RuleHasFired data object for the current timestep to 1.0 with the user-defined RuleHasFiredSlot function in the Priority Functions utility group.

### **A.15.3. SetElVadoReclamationAccount**

*Explanation:* This rule sets the value for the accounting supply to move Reclamation water from El Vado Reservoir to Abiquiu Reservoir.

*Rule Execution:* If the value in the ElVadoReclamation time series slot in the RuleHasFired data object for the current timestep is a NaN as determined with the user-defined RuleHasNOTFired function in the Priority Functions utility group and the result from the user-defined RuleIsInPriority function in the Priority Functions utility group (Refer to Section A.15.1.1) is true, the rule fires.

*Rule Logic:* The user-defined ElVadoReclamationRelease function in the ElVado Account Functions utility group is used to set the value for the supply. If the amount of water in storage at El Vado Reservoir for the Reclamation account at the previous timestep as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group is greater than zero, the function result is set to the minimum of the following three values: 1) the amount of water in storage at El Vado Reservoir for the Reclamation account at the previous timestep as determined with the user-defined PreviousAccountStorage function minus the sum of the supplies from that source as determined with the user-defined SumSupplies function in the Account Summing Functions utility group plus the sum of the supplies into that source as determined with the user-defined SumAllSuppliesIntoSource function in the Account Lists utility group, 2) the available account storage space at the destination reservoir, Lake Abiquiu, as determined with the user-defined AvailableAccountStorage function in the General Account Functions utility group, or 3) the value in the MaximumSJOutflow periodic slot in the ElVadoData data object for the current timestep.

Within the rule, the result is restricted using the predefined Min function to the amount of San Juan water available to be allocated computed as the result from user-defined ReconcileSJOutflow function in the General Account Functions utility group minus the sum of all supplies for San Juan water at El Vado Reservoir.

The rule includes an additional assignment statement to set the value for the ElVadoReclamation time series slot in the RuleHasFired data object for the current



timestep to 1.0 with the user-defined RuleHasFiredSlot function in the Priority Functions utility group.

#### **A.15.4. SetElVadoOtowiPaybacksAccounts**

*Explanation:* This rule sets accounting supplies for contractor deliveries to payback the river at Otowi. The code for this rule matches the code for the SetHeronMRGCDPaybacksAbiquiuAccounts Rule except the identified source reservoir, release type, and destination are different for the supplies to be set. Refer to the discussion for SetHeronMRGCDPaybacksAbiquiuAccounts Rule for more information on how the assignments are made.

#### **A.15.5. SetElVadoAlbuquerquePaybacksAccounts**

*Explanation:* This rule sets supplies for contractors that have debts to Albuquerque to payback Albuquerque at Abiquiu. The code for this rule matches the code for the SetHeronMRGCDPaybacksAbiquiuAccounts Rule except the identified source reservoir, release type, and destination are different for the supplies to be set. Refer to the discussion for the SetHeronMRGCDPaybacksAbiquiuAccounts Rule for more information on how the assignments are made. Also, note that the supplies set with this rule include consideration for the available account storage at Abiquiu as determined with the user-defined AvailableAccountStorage function with consideration for the loss of San Juan water as referenced with the user-defined SJCLoss function.

#### **A.15.6. SetElVadoMRGCDPaybacksAccounts**

*Explanation:* This rule sets supplies for contractors that have debts to MRGCD to payback MRGCD at Abiquiu. The code for this rule matches the code for the SetHeronMRGCDPaybacksAbiquiuAccounts Rule except the identified source reservoir, release type, and destination are different for the supplies to be set. Refer to the discussion for the SetHeronMRGCDPaybacksAbiquiuAccounts Rule for more information on how the assignments are made.

#### **A.15.7. SetElVadoAccountFillAccounts**

*Explanation:* This rule sets supplies for contractors to fill allocated storage space at Abiquiu Reservoir. The code for this rule matches the code for the SetHeronAccountFillAbiquiuAccounts Rule except for the source reservoir, release type, and destination for the supplies to be set. Refer to the discussion for the SetHeronAccountFillAbiquiuAccounts Rule for more information on how the assignments are made.

## **A.16. Preparatory Abiquiu**

The rules in this policy group are used to transfer the portion of native Rio Grande water from El Vado Reservoir that is Emergency Drought water designated for ESA purposed to the Reclamation account at Abiquiu Reservoir. Also, calculations are completed for determining whether native Rio Grande water in storage at Abiquiu Reservoir from flood operations should be locked in as carryover storage until after the irrigation season. Preliminary values for the release of Rio Grande and San Juan-Chama Project water are computed.

### **A.16.1. SetRioGrandeToReclamationTransfer**

*Explanation:* This rule sets the accounting supply to transfer Emergency Drought water designated for ESA purposes received from El Vado Reservoir at Abiquiu Reservoir from the Rio Grande account to the Reclamation account at Abiquiu. Effectively, Emergency Drought water is used to meet targets by filling the Reclamation account at Abiquiu as San Juan-Chama Project water in the Reclamation account is used to meet targets. Note that the Compact calculations are adjusted for this transfer of native Rio Grande water to an account for San Juan-Chama Project water. The transfer is set to the total Rio Grande inflow minus any local inflows to the upstream reach multiplied by the value for the supply for Rio Grande water with a destination of the Reclamation account at Abiquiu divided by the sum of all the supplies for Rio Grande water. The Rio Grande inflow is computed as the total inflow minus the sum of the San Juan supplies with consideration for the loss of San Juan water between El Vado Dam and Abiquiu Reservoir.

*Rule Execution:* If the value for the RioGrandeAbiquiuToReclamationAbiquiu supply for the current timestep is a NaN, the rule fires.

*Rule Logic:* The value for the Supply for the RioGrandeAbiquiuToReclamationAbiquiu transfer is set to the value for the Inflow time series slot in the Abiquiu level power reservoir object for the current timestep minus the sum of all the San Juan supplies from El Vado Reservoir as identified with the user-defined AllSJSuppliesFrom function in the Account Lists utility group and summed with the user-defined SumSuppliesAtDate function in the Account Summing Functions utility group. The calculation includes consideration for the loss of San Juan water as referenced using the SJCLoss function in the Get Data Functions utility group. The value for the Local Inflow time series slot in the ElVadoToAbiquiuLocalInflow reach object for the current timestep is then subtracted. That result is then multiplied by the value for the supply for Rio Grande water with a destination of Reclamation\_Abiquiu divided by the sum of all the supplies for Rio Grande water for all destinations summed using the predefined Sum function. The user-defined AllSuppliesByReleaseTypeDestination function in the Account Lists utility group is used to identify the supplies.

### A.16.2. AbiquiuLockedIn

*Explanation:* This rule is used to set a trigger indicating that storage during flood control operations should be locked in as carryover storage at Abiquiu Reservoir until after the irrigation season per flood control operations. Between July and October, if the storage is greater than the input minimum storage for carryover and the flow at Embudo plus the local inflows above Abiquiu Reservoir are less than an input flow to lockin water as carryover storage, the trigger is set. The water is then released during the non-irrigation season. This rule effectively assures any water stored during flood control operations is delivered to Elephant Butte for the Compact if the flows subsequently decrease to where it would otherwise be diverted in the Middle Valley.

*Rule Execution:* If this rule has not fired as checked against the predefined HasRuleFiredSuccessfully function, the rule fires.

*Rule Logic:* This rule includes two assignment statements. First, the user-defined LockinCarryOverLookAhead function in the Flood Carry Over Functions utility group is used to set the value for the Locked In time series slot in the Abiquiu level power reservoir object if it is a NaN for the current timestep plus the number of timesteps input to the LookAhead table slot in the FloodCarryOverData data object as determined using the user-defined LookAhead function in the Date Functions utility group (hereinafter referred to as the look ahead timestep). The assignment serves as a trigger to identify whether storage should be locked in at Abiquiu Reservoir.

Within the LockinCarryOverLookAhead function, the user-defined IfLockinCarryOverLookAhead function in the Flood Carry Over Functions utility group is used to see if it is time to lock in storage. This function checks first to see if the look ahead date is between July and October (inclusive) as determined with the user-defined FloodCarryOverSeasonLookAhead function in the Date Functions utility group and if the value for the Local Inflow time series slot in the ElVadoLocalInflow reach object plus the value for the Gage Inflow time series slot in the Embudo stream gage object at the look ahead timestep as computed using the user-defined OtowiLookAhead function in the Mainstem Reach Functions utility group is less than the input value in the MinLockinFlow table slot in the FloodCarryOverData data object. Second, this function checks to see if the look ahead date is between July 3<sup>rd</sup> and the end of October (inclusive) and the value for the LockedIn time series slot for any day in this period is equal to 1.0.

An additional check is then included within the LockinCarryOverLookAhead function to see if the result from the user-defined EstimatedRGStorageLookAhead function in the Flood Carry Over Functions utility group is greater than the value for the MinRGCarryOverStorage table slot in the AbiquiuData data object as referenced using the user-defined MinRGCarryOverStorage function in the Flood Carry Over Functions utility group. The EstimatedRGStorageLookAhead function estimates the storage of Rio Grande water at the look ahead date based on the storage at the previous timestep referenced using the PreviousRGStorage function in the General Account Functions utility group and the estimated inflow and outflow till the look ahead date as referenced

using the user-defined EstimatedRGInflowLookAhead and the EstimatedRGOutflowLookAhead functions in the Flood Carry Over Functions utility group. The volume of outflow, till the look ahead date, is assumed to be equal to 3.5 times the Rio Grande outflow at the previous timestep. The estimated inflow is computed using the user-defined EstimatedAbiquiuInflowLookAhead function in the Estimated Inflow Functions utility group as the maximum of either the volume of the Rio Grande outflow from El Vado Dam plus the volume of the Local Inflow time series slot in the ElVadoToAbiquiuLocalInflow reach object or the volume of the Local Inflow time series slot in the ElVadoLocalInflow reach object plus the volume of the Local Inflow time series slot in the ElVadoToAbiquiuLocalInflow reach object till the look ahead dated. That estimated inflow volume is then multiplied by 0.95 and multiplied by the value for the PercentForecastError time series slot in the ForecastData data object for the end of the current month if it is not a NaN.

If those criteria are all satisfied, an interior IF THEN ELSE statement is included in the LockinCarryOverLookAhead function to see if the value for the Locked In time series slot for the timestep before the look ahead timestep is equal to 1.0. If so the function result is 1.0. Otherwise, if the result from the user-defined CochitiFloodSpaceAvailable function in the Cochiti Flood Control Functions utility group is greater than the value in the MinimumFloodSpaceForFloodCarryOver table slot in the CochitiData data object as determined with the user-defined IfCochitiFloodSpaceAvailable function in the Cochiti Flood Control Functions utility group, the function result is 1.0. Otherwise, the function result is 0.0.

A second assignment statement is included in the rule to set the value for the Locked In time series slot for the current timestep using the user-defined LockinCarryOver function in the Flood Carry Over Functions utility group. The logic in this function is the same as for the look ahead timestep except values for the current timestep are used instead of estimated values for the look ahead timestep and the check against the available flood space at Cochiti Lake is not included.

### **A.16.3. ComputeAbiquiuRGRelease**

*Explanation:* This rule computes the preliminary value for the release of Rio Grande water from Abiquiu Dam. The rule includes a separate calculation if conservation storage at Abiquiu Reservoir is being modeled where inflows of native Rio Grande water will be stored as not needed to meet downstream demands and then released as needed to meet target flows in the Middle Valley. Otherwise, the release is set to bypass inflows with any needed release from storage or adjustment from any potential storage during flood control operations and any potential non-irrigation season release of carover storage from the irrigation season.

*Rule Execution:* If the value for the RGOutflow time series slot in the AbiquiuData data object for the current timestep is a NaN, the rule fires.

*Rule Logic:* An IF THEN ELSE statement is used to see if the result from the ComputeAbiquiuRGConsInflow function in the RGConservationFunctions utility group is greater than 0.0 cfs. If not, the value for the RGOutflow time series slot in the AbiquiuData data object for the current timestep is set to the result from the AbiquiuMinRGOutflow function in the Abiquiu Account Functions utility group. If so, an interior IF THEN ELSE statement is used to see if the current timestep is January 1. If so, the RGOutflow slot is set to the result from the AbiquiuMinRGOutflowWRGConsOnJan1 function in the Abiquiu Account Functions utility group. If not, the RGOutflow slot is set to the result from the AbiquiuMinRGOutflow function minus the result from the ComputeAbiquiuRGConsInflow function in the RGConservationFunctions utility group.

The AbiquiuMinRGOutflow function computes the minimum outflow equal to the current Rio Grande inflow as determined with the user-defined CurrentRGInflow function in the General Compute Functions utility group plus the value for the Gain Loss for the Rio Grande Account at Lake Abiquiu for the previous timestep plus the result from the user-defined AbiquiuRGStorageAdjustment function in the Abiquiu Account Functions utility group plus the result from the RGCarryOverRelease function in the Flood Carry Over Functions utility group. The predefined Max function is used to restrict the result from this function to a minimum of the value in the MinRGDemandRelease time series slot in the AbiquiuData data object.

The AbiquiuMinRGOutflowWRGConsOnJan1 function used if the timestep is January 1 computes the RGOutflow in the same manner except the AbiquiuRGStorageAdjustment function is not included and the result is restricted using the predefined Max function to the value for the MinRGFlowToAllowConsStorage periodic slot for the previous timestep, the value in the MinRGOutflow time series slot in the AbiquiuData data object for the current timestep, or the value for the MRGCDDemand time series slot in the AbiquiuData data object for the current timestep.

#### **A.16.4. ComputeAbiquiuMinFlowsDemand**

*Explanation:* This rule determines a computed release of supplemental water needed from Abiquiu Dam required to meet downstream target flows in the Middle Valley. Note that the value is set to zero if there is conservation storage at Cochiti Lake. The value is computed based on the determined total flow needed at Abiquiu minus the release of native Rio Grande water, letter water deliveries, and any release of San Juan-Chama Project water for MRGCD or the Albuquerque surface water diversion. The actual release is set with the SetAbiquiuMinFlowsDemand rule.

*Rule Execution:* If the value for the ComputedMinFlowsDemand time series slot in the AbiquiuData data object for the current timestep is a NaN and the value for the MinFlowsDemand time series slot in the AbiquiuData data object for the next timestep is a NaN, the rule fires.

*Rule Logic:* The value for the ComputedMinFlowsDemand time series slot in the AbiquiuData data object for the current timestep is set to the value for the TotalFlowNeededToMeetTarget time series slot in the AbiquiuData data object for the current timestep referenced with the user-defined TotalFlowAtAbiquiuNeededToMeetMinTarget function in the Abiquiu Account Functions utility group minus the result from the user-defined AbiquiuMRGCDDemandMetWithSJ function in the Abiquiu Account Functions utility group minus the value in the RGOutflow time series slot in the AbiquiuData data object for the current timestep referenced using the user-defined RGOutflow function in the General Account Functions utility group minus the result from the user-defined LetterWaterAdjustmentAbiquiu function in the Demand Functions utility group and minus the delivery to the Albuquerque surface water diversion computed with the user-defined AbiquiuAlbuquerqueDemand function.

An interior IF THEN ELSE statement is included in the equation to see if the result from the user-defined MRGCDAbiquiuSJDemand function in Demand Functions utility group is greater than zero and if the result from the AbiquiuMRGCDDemandMetWithSJ function is within 0.01 cfs of zero as determined using the user-defined IsAbiquiuMRGCDDemandMetWithSJ=0 function in the TargetTestFunctions utility group and if the value for the MRGCDDemand time series slot in the ElVadoData data object for the previous timestep is equal to the value for the RGOutflow time series slot in the ElVadoData data object for the previous timestep as determined using the user-defined IsElVadoMRGCDDemand=ElVadoRGOutflow function in the TargetTestFunctions utility group. If these criteria are met, the MRGCDAbiquiuSJDemand function is used in the equation as opposed to the AbiquiuMRGCDDemandMetWithSJ function.

An exterior IF THEN ELSE statement is used to see if the values in the RGConservationReleaseDays table slot are NaNs and if the current timestep is after the date determined with the user-defined GetStartRGConsReleaseDate function and before the result from the user-defined GetEndRGConsReleaseDate function in the RGConservationFunctions utility group and if the value of the Storage in the RioGrandeConservation account at Abiquiu Reservoir at the day before the result from the GetStartRGConsReleaseDate function is greater than zero. If so, the result from the rule is restricted to the result from the user-defined ConstantRGConservationRelease function in the RGConservationFunctions utility group using the predefined MaxItem function.

#### **A.16.5. SetAbiquiuMinFlowsDemand**

*Explanation:* This rule sets the value for the minimum flow needed to meet downstream target flows for the current timestep with reference to the computed need. The actual release is not adjusted unless a specific threshold change in the need is exceeded and the release of supplemental water is not adjusted until after an input minimum of days since the last adjustment. The calculations are included to better represent actual operations

that do not entail adjusting the release each day by a few cfs in attempt to exactly meet a target flow.

*Rule Execution:* If the value for the MinFlowsDemand time series slot in the AbiquiuData data object for the next timestep is a NaN, the rule fires.

*Rule Logic:* An IF THEN ELSE statement is included to first check if the current timestep is the Start Timestep. If so, the value for the MinFlowsDemand time series slot in the AbiquiuData data object for the current timestep is set to the value in the ComputedMinFlowsDemand time series slot for the current timestep rounded to the nearest 10 cfs. A separate assignment is included to reset a counter for the number of days since an adjustment was made to the release of supplemental water.

Otherwise, another IF THEN ELSE statement is included to check if the previous counter value in the CountSinceMinFlowsDemandAdjusted time series slot is greater than or equal to the value in the MinDaysBeforeMinFlowsDemandAdjusted scalar slot in the Abiquiu Data data object and the average demand over that period minus the previous value for the MinFlowsDemand time series slot in the AbiquiuData data object exceeds the value for the ChangeBeforeMinFlowsDemandAdjusted scalar slot in the AbiquiuData data object or the previous value in the CountSinceMinFlowsDemandAdjusted time series slot in the AbiquiuData data object is greater than or equal to the value in the MaxDaysWithNoMinFlowsDemandAdjusted scalar slot in the AbiquiuData data object. If the condition is satisfied, the value for the MinFlowsDemand slot is set to the average of the ComputedMinFlowsDemand slot values since the last adjustment was made. Otherwise, the previous value for the MinFlowsDemand slot is maintained. A counter is also set if the value in the MinFlowsDemand slot is adjusted.

#### **A.16.6. SetAlbuquerqueLoanAccounts**

*Explanation:* This rule sets the values for supplies for Albuquerque loans to other contractors based on input priorities if Albuquerque has the water available to loan based on an account storage that exceeds an input threshold before Albuquerque will loan water and whether the borrowed water can be paid back in the same year. The loans are limited based on an input maximum.

*Rule Execution:* If the value for the Supply with a release type of Albuquerque loan, source of the Albuquerque account at Abiquiu Reservoir and a destination of Reclamation\_Abiquiu as identified with the user-defined SuppliesIntraBySourcesReleaseTypeDestinations function in the Account Lists utility group is a NaN, the rule fires.

*Rule Logic:* A FOREACH loop is used to set the supply for Albuquerque to loan water to accounts at Abiquiu Reservoir, other than the Albuquerque account, with a release type of OtowiPaybacks and destination of Otowi\_Otowi. The list is generated with the user-defined PrioritizedAccounts function in the Priority Functions utility group which creates

a list of accounts with the corresponding priority as input to the AccountReleasePriority periodic slot in the AbiquiuData data object. The predefined SortPairsAscending function is used to sort the list based on the input priorities, and the user-defined AlbuquerqueAbiquiuEXAccounts function in the Account Lists utility group is referenced to identify the accounts to be sorted.

The user-defined GetSortedLoanAccounts function in the Account Lists utility group is used to create an array with the supply and computed loan amount. The user-defined SuppliesIntraBySourcesReleaseTypeDestinations function in the Account Lists utility group is used to identify the upstream supplies to be set as those with a release type of AlbuquerqueLoan, destination of Abiquiu Reservoir, and source as the Albuquerque account at Lake Abiquiu. The user-defined SuppliesBySourcesReleaseTypeDestination function is used to create a list of supplies for the same accounts that have a release type of OtowiPaybacks and destination of Otowi\_Otowi. Those supplies are referenced to determine the current debt for each of those accounts. An exterior IF THEN ELSE statement is used to see if the account can borrow the maximum amount of water from the Albuquerque account and pay it back in the same year. If not, the loan is set to the current debt as determined with the user-defined GetAccountDebt function in the Exchange Functions utility group minus the Storage for the account at Abiquiu Reservoir at the previous timestep as referenced using the user-defined PreviousAccountStorage function in the General Account Functions utility group. That result is restricted to be no greater than the input value to the MaxDeliveryRequestRelease table slot in the AbiquiuData data object referenced using the user-defined MaxDeliveryRequestRelease function in the Get Data Functions utility group minus the tally of the loans as tracked in the function. If the account can borrow the water and pay it back in the same year, the loan is set to the maximum as determined with the user-defined MaximumAlbuquerqueLoan function in the Abiquiu Account Functions utility group, again, restricted to be no greater than the input maximum delivery minus the tally of the loans.

Three separate assignment statements are also included in the rule to set supplies for Albuquerque to loan water to the Reclamation, NMISC, and MRGCD accounts at Abiquiu Reservoir. The user-defined AlbuquerqueWillLoanToAccount function in the Exchange Functions utility group is used to check the value for the AlbuquerqueLoanSwitch periodic slot in the AbiquiuData data object for each of the three accounts. If the switch is equal to 1.0, a loan amount is computed.

### **A.16.7. ComputeAbiquiuSJRelease**

*Explanation:* This rule computes a preliminary value for the release of San Juan water from Abiquiu Dam. The release is essentially set to any release of MRGCD San Juan-Chama Project needed to meet the MRGCD demand (MRGCD San Juan-Chama Project water is used after native supplies are exhausted) plus any release from the Reclamation account to meet target flows in the Middle Valley plus any release of Albuquerque water for the surface water diversion plus any letter water deliveries plus any other passthrough



water at Abiquiu which may include deliveries to the Cochiti rec pool. The total is checked against an input maximum release of San Juan-Chama Project water.

*Rule Execution:* If the value for the SJOutflow time series slot in the AbiquiuData data object for the current timestep is a NaN, the rule fires.

*Rule Logic:* This function includes an IF THEN ELSE statement to see if it is the irrigation season (March 1 to October 31) as determined with the user-defined IrrigationSeason function in the Date Functions utility group and if the value for the MRGCDDemand time series slot in the AbiquiuData data object for the current timestep referenced with the user-defined AbiquiuMRGCDDemand function in the Abiquiu Account Functions utility group is less than the value in the MinRelease periodic slot in the AbiquiuData data object for the current timestep referenced using the MinOutflow function in the Get Data Functions utility group or less than the result from the user-defined AbiquiuMinRGOutflow function in the Abiquiu Account Functions utility group. If not, the value for the SJOutflow time series slot in the AbiquiuData data object for the current timestep is set to the result from the AbiquiuSJOutflow function in the Abiquiu Account Functions utility group. That function references the user-defined TotalPotentialDestinationReleaseAbiquiu function in the General Compute Functions utility group, and the result is restricted to a maximum of the value in the MaximumSJOutflow periodic slot in the AbiquiuData data object for the current timestep. Otherwise, the value for the SJOutflow slot is set to sum of all the supplies below El Vado Dam with consideration for the loss of San Juan water as referenced using the user-defined SJCLoss function in the Get Data Functions utility group plus the letter water adjustment as determined with the user-defined LetterWaterAdjustmentAbiquiu function in the Demand Functions utility group plus the San Juan water to be released by Reclamation to meet downstream target flows as determined with the AbiquiuMinFlowsSJRelease function in the Abiquiu Account Functions utility group plus the result from the user-defined AbiquiuAlbuquerqueDemand function in the Abiquiu Account Functions utility group.

## **A.17. Abiquiu**

The rules in this policy group are used to determine the total outflow from Abiquiu Dam with consideration for minimum flows, pre-evacuation releases, stepped release restrictions, downstream channel capacity restrictions, prescribed maintenance flows, and flood control operations.

### **A.17.1. Abiquiu Outflow**

*Explanation:* Unless a total outflow for Abiquiu Dam has been input for the current timestep, the value for the outflow is set to the determined Rio Grande release plus the release of San Juan-Chama Project water. The resulting total release is checked to see if

it is physically legitimate based on the outlet works, and if not, the release is reset to reflect the restriction for the given outlet works.

*Rule Execution:* If the value for the Outflow time series slot in the Abiquiu level power reservoir object for the current timestep is a NaN, the rule fires.

*Rule Logic:* The user-defined InitialAbiquiuOutflow function in the Abiquiu Account Functions utility group is used to compute the total outflow from Abiquiu Dam. This function contains an exterior IF THEN ELSE statement to see if the value in the TotalOutflow time series slot in the AbiquiuData data object for the current timestep as referenced using the user-defined TotalOutflow function in the Get Data Functions utility group is a NaN. If not, the value for the Outflow time series slot in the Abiquiu level power reservoir object for the current timestep is set to that input TotalOutflow value. Otherwise, a second IF THEN ELSE statement is used to see if it is the irrigation season as determined with the user-defined IrrigationSeason function and whether the stipulations of Article VII of the Compact are NOT in effect as determined with the user-defined CompactVIIinEffect function. If not, the result is set to the sum of the values for the SJOutflow and RGOutflow time series slots in the AbiquiuData data object for the current timestep as referenced with the user-defined SJOutflow and RGOutflow functions in the General Account Function utility group and restricted to the value in the MinRelease periodic slot in the AbiquiuData data object for the current timestep as referenced using the MinOutflow function in the Get Data Functions utility group. If so, a third IF THEN ELSE statement is used to see if the value in the Outflow time series slot in the El Vado storage reservoir object for the previous timestep minus the sum of the supplies at El Vado Reservoir with a destination of Abiquiu Reservoir is greater than or equal to the value in the MRGCDDemand time series slot in the ElVadoData data object for the previous timestep. If not, the release is again computed as the sum of the RGOutflow and SJOutflow slot values but is restricted to assure it does not exceed the result from the user-defined AbiquiuMiddleValleyDemand function in the Get Data Functions utility group. A fourth IF THEN ELSE statement is used to see if the value in the Outflow time series slot in the El Vado storage reservoir object for the previous timestep is greater than the supply to transfer water from the Indian Storage account at El Vado Reservoir downstream for the previous timestep. If not, the Outflow slot is set to the maximum of either the result from the user-defined AbiquiuMiddleValleyDemand function or the value for the RGOutflow time series slot for the current timestep plus the sum of the supplies from El Vado Reservoir with destinations of CochitiRecPool\_Cochiti, Jemez\_Jemez, or Albuquerque\_ElephantButte with consideration for the loss of San Juan water as referenced with the user-defined SJCLoss function plus the San Juan water to be released by Reclamation to meet downstream target flows as determined with the AbiquiuMinFlowsSJRelease function in the Abiquiu Account Functions utility group plus the result from the user-defined AlbuquerqueMVDemand function in the Demand Functions utility group. That result is also restricted to the value in the MinRelease periodic slot in the AbiquiuData data object. If the last criterion is not satisfied, the equation is the same except the letter water adjustment is added.

Within the rule, the result for the Outflow is checked against the physical restrictions of the outlet works using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1).

### **A.17.2. AbiquiuMinimumFlow**

*Explanation:* The total outflow from Abiquiu Dam is reset to an input minimum if the computed initial value does not exceed either the input minimum release or input minimum Rio Grande outflow.

*Rule Execution:* The rule fires if the value for the Outflow time series slot in the Abiquiu level power reservoir object for the current timestep is less than or equal to the value in the MinRelease periodic slot in the AbiquiuData data object for the current timestep referenced using the MinOutflow function in the Get Data Functions utility group or less than or equal to the value in the MinRGOutflow periodic slot in the AbiquiuData data object referenced with the user-defined MinRGOutflow function in the Get Data Functions utility group AND the rule has not fired as checked against the predefined HasRuleFiredSuccessfully function.

*Rule Logic:* If the execution constraints are satisfied, the value in the Outflow time series slot in the Abiquiu level power reservoir object for the current timestep is reset to the maximum of the value in the MinRelease periodic slot in the AbiquiuData data object for the current timestep referenced with the MinOutflow function or the value in the MinRGOutflow periodic slot in the AbiquiuData data object referenced using the user-defined MinRGOutflow function. The result is checked against the physical restrictions of the outlet works using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1). A PRINT statement is included to print a comment to the Diagnostics Output window that the computed outflow was less than the input minimum.

### **A.17.3. AbiquiuPreEvacuation**

*Explanation:* If water needs to be pre-evacuated at Abiquiu Reservoir in anticipation of forecasted inflows, the outflow from Abiquiu Dam is reset based on a computed pre-evacuation flow which includes consideration for the downstream channel capacity and stepped release restrictions.

*Rule Execution:* The rule fires if the result from the user-defined IsPreEvacuationRequired function in the AbiquiuPreEvacuationFunctions utility group is TRUE AND the rule has not fired as checked against the predefined HasRuleFiredSuccessfully function. The IsPreEvacuationRequired function checks to see if the current timestep is between December 31 plus the number of days entered to the PreEvacJulianDay column of the PreEvacuationData table slot in the AbiquiuData data object and April 30 (inclusive) as determined with the user-defined IsPreEvacuationSeason function in the AbiquiuPreEvacuationFunctions utility group.

The IsPreEvacuationRequired function also checks to see if the forecasted volume of inflows to Abiquiu Reservoir minus the space remaining to the maximum elevation at El Vado Reservoir minus the space remaining to the AllowableStorageElevation entered to the PreEvacuationData table slot as computed with the user-defined AbiquiuRemainingForecastVolume function is greater than or equal to the input value in the MinPreEvacStorage column in the PreEvacuationData table slot.

*Rule Logic:* If the execution constraints are satisfied, the value in the Outflow time series slot in the Abiquiu level power reservoir object for the current timestep is reset to the result from the user-defined DeterminePreEvacFlow function in the AbiquiuPreEvacuationFunctions utility group. The user-defined ComputePreEvacFlow function in that utility group is referenced to compute the pre evacuation flow rate as the average flow required to evacuate the result from the AbiquiuRemainingForecastVolume function over the period from the current timestep to August 1. Within the DeterminePreEvacFlow function, the result is checked against the input downstream (D/S) value in the ChannelCapacities table slot and input value for the MaxThreshold, MaxStepFactor, and MaxThreshold in the SteppedReleaseData table slot in the AbiquiuData data object.

The result from the DeterminePreEvacFlow function is checked against the physical restrictions of the outlet works using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1). Refer to Figure A.44 for a flowchart that depicts how preevacuation releases are set.

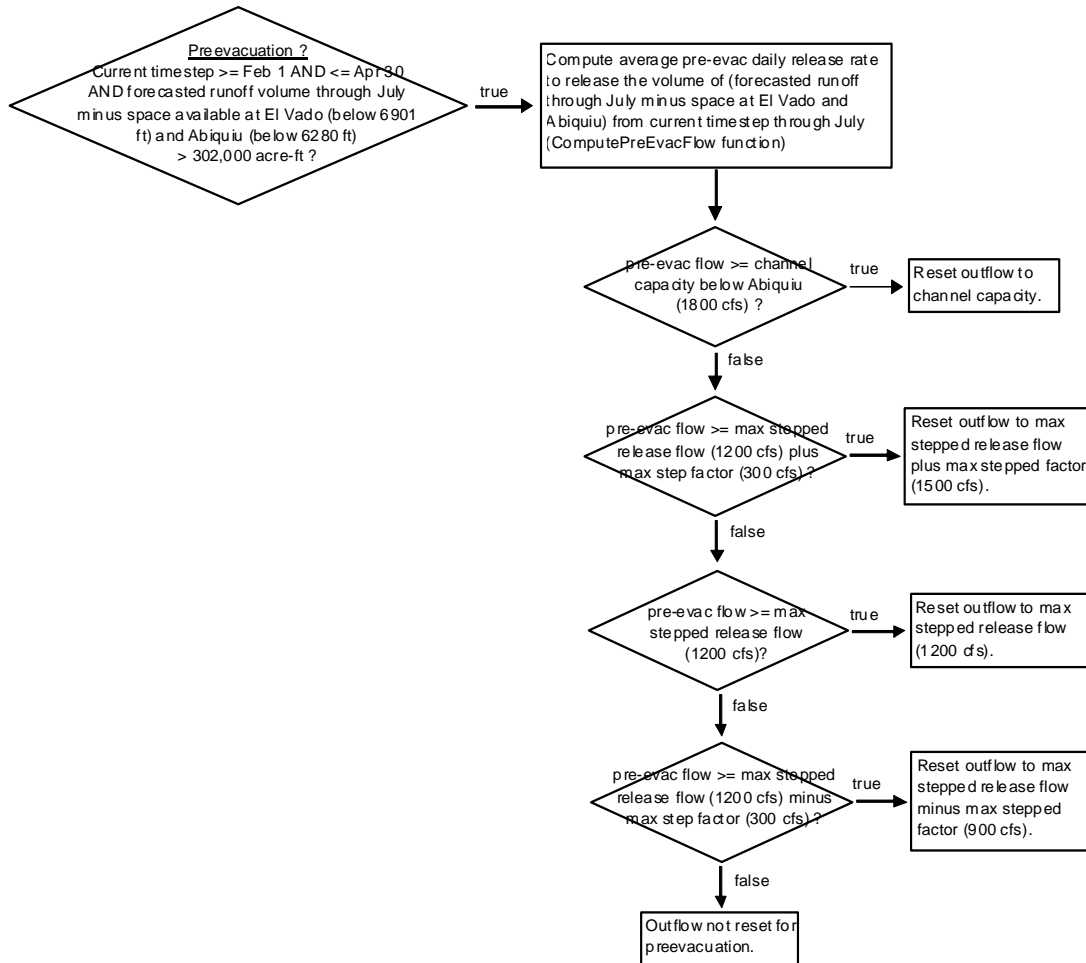


Figure A.44. Flow Chart with Logic for Setting Abiquiu Preevacuation Releases

#### A.17.4. AbiquiuSteppedRelease

*Explanation:* Per flood control operations, the total outflow from Abiquiu Dam may be reset if the release needs to be stepped. This rule determines whether a stepped release is required, and if so, computes the release based on a stepped release approach.

*Rule Execution:* The rule fires if the rule has not fired as checked against the predefined HasRuleFiredSuccessfully function and if the result from the user-defined SteppedReleaseIsNeeded function in the Stepped Release Functions utility group is TRUE. The SteppedReleaseIsNeeded function includes three separate functions to check the change to the outflow as computed using the user-defined ComputeDeltaOutflow function in the General Compute Functions utility group against different criteria.

*Rule Logic:* If the execution constraints are satisfied, the value in the Outflow time series slot in the Abiquiu level power reservoir object for the current timestep is reset to the result from the user-defined DetermineSteppedRelease function in the Stepped Release

Functions utility group. This function includes an exterior IF THEN ELSE statement to see if the outflow from Abiquiu Dam is increasing as determined using the user-defined OutflowIsIncreasing function in the General Boolean Functions utility group. If so, the outflow is calculated with the user-defined ComputeOutflowIncrSteppedRelease function. If not, the outflow is computed using the user-defined ComputeOutflowDecrSteppedRelease function in the Stepped Release Functions utility group unless it is determined that the release should be shutoff.

Within the rule, the result from the DetermineSteppedRelease function is checked against the physical restrictions of the outlet works using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1). Refer to Figure A.45 through A.47 for flowcharts that show how stepped releases are set.

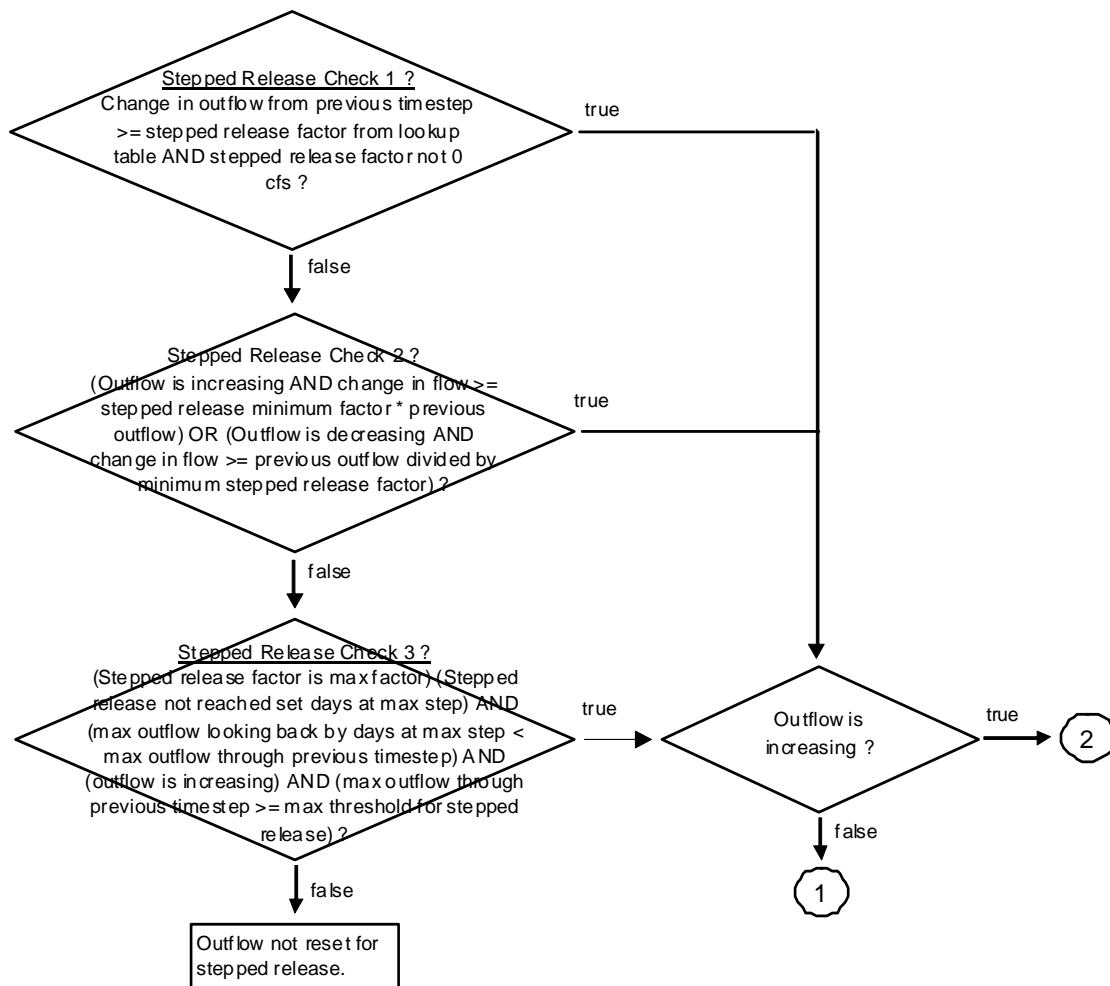


Figure A.45. Flow Chart Depicting Checks for whether Stepped Release Needed

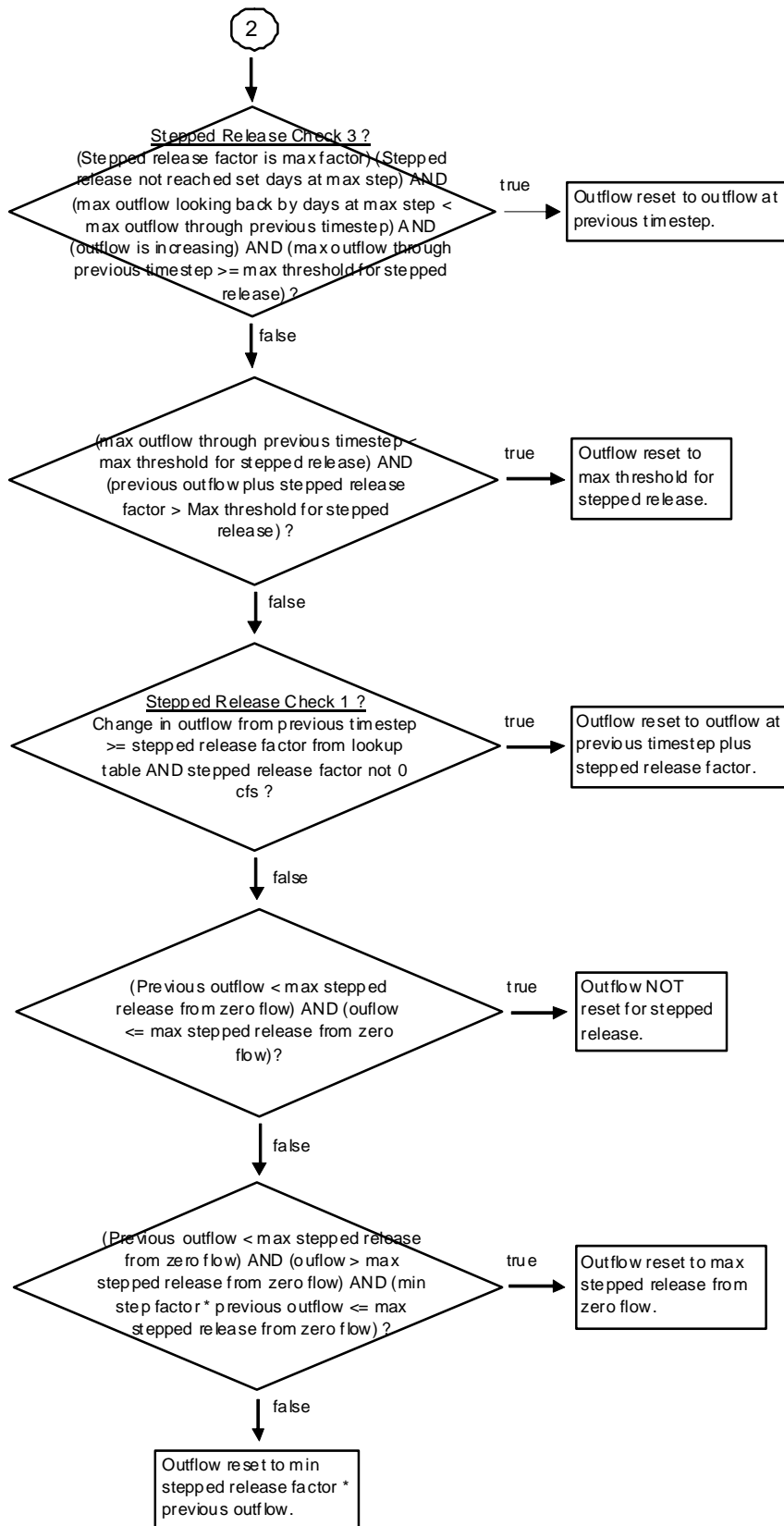


Figure A.46. Flow Chart for Calculation of Stepped Release when Outflow is Increasing

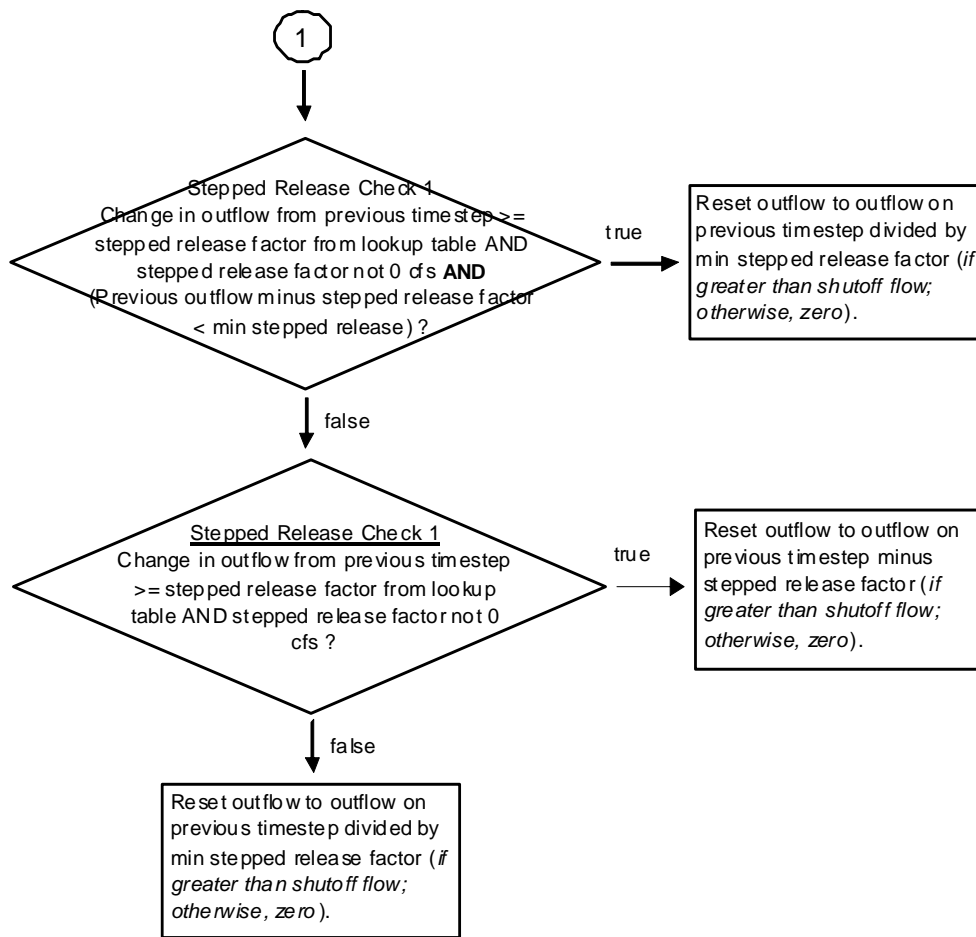


Figure A.47. Flow Chart for Calculation of Stepped Release when Outflow is Decreasing

### A.17.5. AbiquiuChannelCapacityRestrictions

**Explanation:** This rule resets the outflow from Abiquiu Dam if necessary maintain a downstream flow that is less than the downstream channel capacity restrictions below the dam (1800 cfs), at Chamita (3000 cfs), and at Otowi (10,000 cfs) with consideration for downstream local inflows.

**Rule Execution:** This rule fires if the rule has not fired as checked against the predefined HasRuleFiredSuccessfully function.

**Rule Logic:** This rule includes an IF THEN ELSE statement to see if the value for the Outflow time series slot in the Abiquiu level power reservoir object for the current timestep is greater than the maximum release to comply with downstream channel capacities at Otowi and Chamita as determined with the user-defined AbiquiuReleaseForChannelCapacity function in the Abiquiu Channel Capacity Functions utility group. This function references input values in the ChannelCapacities table slot in



the AbiquiuData data object along with the flow at Otowi and local inflows between Embudo and Otowi and local inflows between Abiquiu Dam and Chamita to determine the maximum release. The value for the Outflow slot is reset if needed to comply with the downstream channel capacities. The result is checked against the physical restrictions of the outlet works using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1).

### **A.17.6. AbiquiuTemporaryFlowsForMaintenance**

*Explanation:* This rule resets the outflow from Abiquiu Dam to the input maintenance flow if the maintenance switch is set.

*Rule Execution:* The rule fires if the value for the MaintenanceSwitch periodic slot in the AbiquiuData data object is equal to 1.0 as determined with the user-defined IsMaintenanceSwitchOn function in the Maintenance Functions utility group and the rule has not fired as checked against the predefined HasRuleFiredSuccessfully function.

*Rule Logic:* If the execution constraints are satisfied, the value for the Outflow time series slot in the Abiquiu level power reservoir object for the current timestep is reset to the input value in the MaintenanceFlow table slot in the AbiquiuData data object as referenced with the user-defined MaintenanceFlow function in the Maintenance Functions utility group.

### **A.17.7. AbiquiuFloodControl**

*Explanation:* If there is an unregulated spill, per flood control operations, this rule resets the outflow from Abiquiu Dam to either the downstream channel capacity or the unregulated spill if it is higher.

*Rule Execution:* The rule fires if the rule has not fired as checked against the predefined HasRuleFiredSuccessfully function and the value for the Unregulated Spill time series slot in the Abiquiu level power reservoir object is greater than zero as determined with the user-defined ReservoirIsSpillingUnreg function in the General Boolean Functions utility group.

*Rule Logic:* If the execution constraints are satisfied, the value for the Outflow time series slot in the Abiquiu level power reservoir object for the current timestep is reset to the result from the AbiquiuFCOutflow function in the Abiquiu Flood Control Functions utility group. This function includes an IF THEN ELSE statement to see if the result from the user-defined UnregulatedSpillWhenNoConduitFlow function, which references the predefined GetMinSpillGivenInflowRelease function, is less than the value for the downstream channel capacity in the ChannelCapacities table slot in the AbiquiuData data object. If so, the outflow is set to that input downstream channel capacity. If not, the release is set to the result from the UnregulatedSpillWhenNoConduitFlow function.

Within the rule, the result is checked against the physical restrictions of the outlet works using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1).

### **A.17.8. AbiquiuRGCarryOver**

*Explanation:* This rule records the value for any release of carryover storage (during the period from November through March) and updates the recorded value for the carryover storage remaining. Carryover storage results from storage during flood control operations that is not evacuated before flows decrease below a set threshold.

*Rule Execution:* The rule fires if the value for the RGCarryOverRelease time series slot in the AbiquiuData data object for the current timestep is a NaN and the current timestep is between November and March (inclusive) as determined with the user-defined FloodCarryOverReleaseSeason function in the Date Functions utility group.

*Rule Logic:* The rule includes an IF THEN ELSE statement to see if the current timestep is January 1 and if the result from the user-defined IsRGConservationAllowed function in the RGConservationFunctions utility group is TRUE. If not, the value for the RGCarryOverRelease time series slot in the AbiquiuData data object for the current timestep is set to the result from the user-defined RGCarryOverRelease function in the Flood Carry Over Functions utility group. If so, an interior IF THEN ELSE statement is used to see if the value for the RGCarryOverLeft time series slot in the AbiquiuData data object for the previous timestep is greater than the result from the user-defined ComputeAbiquiuRGConsInflow function in the RGConservationFunctions utility group converted to a volume using the predefined FlowToVolume function. If so, the value for the RGCarryOverRelease time series slot for the current timestep is set to the result from the user-defined RGCarryOverRelease function. If not, the value for the RGCarryOverRelease slot is set to 0.0 cfs.

This rule includes an additional assignment statement to set the value for the RGCarryOverLeft time series slot in the AbiquiuData data object for the current timestep to the result from the user-defined RGCarryOverLeft function in the Flood Carry Over Functions utility group.

### **A.18. Abiquiu Accounts**

The rules in this policy group are used to set the accounting supplies for the transfer of Rio Grande water to conservation storage at Abiquiu, release of Rio Grande water, and releases of San Juan-Chama Project water (and Rio Grande conservation storage) from Abiquiu Dam to downstream destinations or payback debts.

### **A.18.1. SetAbiquiuRGConservationAccount**

*Explanation:* This rule sets the supply to transfer water from the Rio Grande account at Abiquiu Reservoir to the Rio Grande Conservation account if conservation storage is being modeled.

*Rule Execution:* If the value for the supply to transfer water from the Rio Grande account at Abiquiu Reservoir to the RioGrandeConservation account as identified with the user-defined AllSuppliesIntraBySourcesReleaseTypeDestination function in the Account Lists utility group is a NaN, the rule fires.

*Rule Logic:* The value for the supply to transfer water from the Rio Grande account at Abiquiu Reservoir to the RioGrandeConservation account as identified with the user-defined AllSuppliesIntraBySourcesReleaseTypeDestination function in the Account Lists utility group is set with the user-defined ComputeAbiquiuRGConsInflow function in the RGConservationFunctions utility group. This function includes an exterior IF THEN ELSE statement to see if the result from the user-defined IsRGConservationAllowed function in the RGConservationFunctions utility group is TRUE. If not, the value for the supply is set to 0.0 cfs. Water will only be transferred to the Rio Grande Conservation account if several criteria are satisfied as included in the IsRGConservationAllowed function such as the following: the Rio Grande inflow is sufficient to meet the MRGCD demand, there is conservation storage space available, the stipulations of Article VII of the Compact are NOT in effect, the Rio Grande inflow is sufficient to meet downstream demands. If so, a second IF THEN ELSE statement is used to see if the current timestep is January 1 and if the value in the Incidental Content time series slot for the previous timestep is greater than 0.0 acre-ft. If so, the supply is set to the current Rio Grande inflow as determined with the user-defined CurrentRGInflow function in the General Compute Functions utility group plus the Gain Loss for the Rio Grande account at the previous timestep as calculated with the user-defined RGConsInflowWPrevGainLoss function in the RGConservationFunctions utility group plus the volume of the Incidental Content time series slot for the previous timestep plus the result from the user-defined RGCarryOverRelease function in the Flood Carry Over Functions utility group minus the result from the AbiquiuRGOutflowWRGConsTransfer function in the RGConservationFunctions utility group. That result is restricted to be no greater than the minimum of either the result from the user-defined AbiquiuEasementSpaceAvailableAsFlow function or the AbiquiuRGConsSpaceAvailableAsFlow function in the RGConservationFunction utility group. If the criteria for the interior IF THEN ELSE statement are not satisfied, the equation is the same except the result for the user-defined AbiquiuRGStorageAdjustment function in the Abiquiu Account Functions utility group is referenced as opposed to the value for the Incidental Content slot.

### A.18.2. SetAbiquiuRGAccount

*Explanation:* This rule sets the accounting supply for the release of Rio Grande water from Abiquiu Dam.

*Rule Execution:* If the value for the supply to transfer water from the Rio Grande account at Abiquiu Reservoir to the destination of RioGrande\_RioGrande as identified with the user-defined SuppliesByReleaseTypeDestination function in the Account Lists utility group is a NaN, the rule fires.

*Rule Logic:* The value for the supply to transfer water from the Rio Grande account at Abiquiu Reservoir to the destination of RioGrande\_RioGrande as identified with the user-defined SuppliesByReleaseTypeDestination function in the Account Lists utility group is set using the user-defined ReconcileRGOutflow function in the General Account Functions utility group. The equation in this function for Abiquiu Reservoir includes an IF THEN ELSE statement to see if the value for the RGOutflow time series slot in the AbiquiuData data object for the current timestep as referenced with the user-defined RGOutflow function in the General Account Functions utility group is greater than the value in the Outflow time series slot in the Abiquiu level power reservoir object. If so, the function result is set to that Outflow for the Abiquiu level power reservoir object. Otherwise, a second IF THEN ELSE statement is used to see if the value for the Outflow time series slot for the Abiquiu level power reservoir object is greater than the sum of the values in the RGOutflow and SJOutflow time series slots in the AbiquiuData data object. If not, the result is set to the value in the RGOutflow time series slot. If so, a third IF THEN ELSE statement is used to see if it is the irrigation season (i.e. the current timestep is between March 1 and October 31 (inclusive)) as determined using the user-defined IrrigationSeason function in the Date Functions utility group and the Storage in the Rio Grande account at the previous timestep is greater than 0.0 acre-ft. If those criteria are NOT satisfied, the function result is set to the value for the Outflow time series slot for the Abiquiu level power reservoir object minus the SJOutflow time series slot for the current timestep restricted to be no less than the value in the MinRGOutflow periodic slot in the AbiquiuData data object referenced using the user-defined MinRGOutflow function in the Get Data Functions utility group. If so, the function result is set to the outflow from the Abiquiu level power reservoir object minus the quantity of the value of the SJOutflow time series slot minus the quantity of the Outflow from the Abiquiu level power reservoir object minus the quantity of the value for the RGOutflow time series slot for the current timestep plus the value of the SJOutflow time series slot all for the current timestep. That result is also restricted to be no less than the value in the MinRGOutflow periodic slot in the AbiquiuData data object.

### **A.18.3. CheckAbiquiuRGOutflow**

*Explanation:* Aborts the simulation if the supply for release of Rio Grande water from the Abiquiu Dam was not set. This rule helps with debugging because a simulation will stop if the Rio Grande supply was not set and the problem and any missing input can be identified by backtracking in the calculations from the point in the simulation that the run aborted.

*Rule Execution:* There are no execution constraints for this rule.

*Rule Logic:* If the value for the Supply at Abiquiu Reservoir with a release type of Rio Grande and destination of RioGrande\_RioGrande is a NaN, the STOP\_RUN command is used to stop the simulation, and the PRINT statement is used to print a comment to the Diagnostics Output window that the run stopped because the account slot value was not set.

### **A.18.4. SetAbiquiuFlowThroughAccounts**

*Explanation:* This rule sets all the accounting supplies for the flowthrough accounts at Abiquiu Reservoir.

*Rule Execution:* If the value in the AbiquiuFlowThrough time series slot in the RuleHasFired data object for the current timestep is a NaN as determined with the user-defined RuleHasNOTFired function in the Priority Functions utility group and the result from the user-defined RuleIsInPriority function in the Priority Functions utility group (Refer to Section A.12.1.1) is TRUE, the rule fires.

*Rule Logic:* The user-defined GetSortedFlowThruAccounts function in the Account Lists utility group is used to create a list of supplies and corresponding values to be set. The values are then set in the rule using a FOREACH loop. Within the function, a WITH DO loop is used with the predefined AccountNamesFromObjReleaseDestination function to create a list of accounts at Abiquiu Reservoir with a release type of FlowThrough. A second WITH DO loop is used to create a list of the Supplies to be set for all those accounts which have a release type of FlowThrough. An array is then created with the name for each supply and the corresponding supply value. If water is available, values for the supplies are set. A third WITH DO loop is used to identify an upstream slot name to be referenced in the upstream object, and a fourth WITH DO loop is used to identify the upstream reservoir. The value for each supply is set as follows: first, a list of supplies is created from the previously generated list of accounts using the user-defined AllSuppliesIntoBySourcesReleaseTypeDestination function in the Account Lists utility group with the designated source as the upstream object. The values for those supplies at the timestep equal to the current timestep minus the number of timesteps input to the Lags table slot in the SanJuanChamaRules data object as referenced with the user-defined

SJCLags function in the Get Data Functions utility group are summed utilizing the user-defined SumSuppliesAtDate function in the Account Summing Functions utility group. The loss of San Juan water between the upstream reservoir and Abiquiu Reservoir as input to the Losses table slot in the SanJuanChamaRules data object is applied to the previous summation. Finally, the water in storage at Abiquiu Reservoir for the account at the previous timestep as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group is added. The predefined Min function is used to restrict that final result to the amount of San Juan water remaining to be allocated as tallied within the function. The available water is computed in the rule as the result from user-defined ReconcileSJOutflow function in the General Account Functions utility group minus the sum of all supplies for San Juan water at Abiquiu Reservoir plus the sum of the supplies from the RioGrandeConservation account at Abiquiu Reservoir.

The rule includes an additional assignment statement to set the value for the AbiquiuFlowThrough time series slot in the RuleHasFired data object for the current timestep to 1.0 using the user-defined RuleHasFiredSlot function in the Priority Functions utility group.

#### **A.18.5. SetAbiquiuOtowipaybacksAccounts**

*Explanation:* This rule sets the accounting supplies for contractor releases of San Juan-Chama Project water as letter water deliveries to payback the river at Otowi.

*Rule Execution:* If the value in the AbiquiuOtowipaybacks time series slot in the RuleHasFired data object for the current timestep is a NaN as determined with the user-defined RuleHasNOTFired function in the Priority Functions utility group and the result from the user-defined RuleIsInPriority function in the Priority Functions utility group (Refer to Section A.12.1.1) is TRUE, the rule fires.

*Rule Logic:* The user-defined GetSortedPaybackAccounts function in the Account Lists utility group is used to create a list of supplies and corresponding values to be set. The values are then set in the rule using a FOREACH loop. Within the function, a WITH DO loop is used with the user-defined PrioritizedAccounts function in the Priority Functions utility group to create a list of accounts with the corresponding priority as input to the AccountReleasePriority periodic slot in the AbiquiuData data object. The predefined SortPairsAscending function is used to sort the list based on the priorities. A second WITH DO loop is used to create a list of the Supplies to be set for all the accounts in that list which have a release type of Otowipaybacks and destination of Otowi\_Otow. An array is then created with the name for each supply and the corresponding payback debt. If water is available to set the Supplies to payback debt, the supply is set to the minimum of the following three values: the value for the debt slot for the payback source as determined with the user-defined GetAccountDebt function in the Exchange Functions utility group which references the predefined GetPaybackDebt function, the amount of San Juan water remaining to be allocated as tallied within the function, or the amount of

water in storage at Abiquiu Reservoir for the account as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group minus the sum of the supplies from that source as determined with the user-defined SumSupplies function in the Account Summing Function utility group plus the sum of the supplies into that source as determined with the user-defined SumAllSuppliesIntoSource function in the Account Lists utility group.

This rule includes an additional assignment statement to set the value for the AbiquiuOtwiPaybacks time series slot in the RuleHasFired data object for the current timestep to 1.0 using the user-defined RuleHasFiredSlot function in the Priority Functions utility group.

### **A.18.6. SetAbiquiuMRGCDAccount**

*Explanation:* This rule sets the accounting supply for the release of MRGCD San Juan-Chama Project water to meet their demand in the Middle Valley.

*Rule Execution:* If the value in the AbiquiuMRGCD time series slot in the RuleHasFired data object for the current timestep is a NaN as determined with the user-defined RuleHasNOTFired function in the Priority Functions utility group and the result from the user-defined RuleIsInPriority function in the Priority Functions utility group (Refer to Section A.12.1.1) is TRUE, the rule fires.

*Rule Logic:* The value for the supply with a release type of MRGCD and destination of MRGCD\_MiddleValley as identified with the user-defined SuppliesByReleaseTypeDestination function in the Account Lists utility group is set to the minimum of the following three computed values: 1) the water in storage at Abiquiu Reservoir for the MRGCD account at the previous timestep as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group minus the sum of the supplies from that source as identified with the user-defined AllSuppliesFromSource function in the Account Lists utility group and summed using the user-defined SumSupplies function in the Account Summing Functions utility group plus the sum of the supplies at Abiquiu Reservoir with a destination of MRGCD\_Abiquiu as identified with the user-defined AllSuppliesIntraByReleaseTypeDestination function in the Account Lists utility group and summed using the user-defined SumSuppliesAtDate function in the Account Summing Functions utility group plus the sum of all the supplies at El Vado Reservoir with a destination of MRGCD\_Abiquiu summed utilizing the user-defined SumSuppliesAtDate function and including consideration for the loss of the San Juan water from El Vado Dam to Abiquiu Reservoir as referenced using the user-defined SJCLoss function in the Get Data Functions utility group, 2) the value for the MRGCDDemand time series slot in the AbiquiuData data object for the current timestep minus the supply at Abiquiu Reservoir with a release type of Rio Grande and destination of RioGrande\_RioGrande minus the result from the user-defined LetterWaterAdjustmentAbiquiu function in the Demand Functions utility group, or 3) the amount water available computed initially in a WITH DO loop as the result

from user-defined ReconcileSJOutflow function in the General Account Functions utility group minus the sum of all supplies for San Juan water at Abiquiu Reservoir minus the sum of the supplies from the RioGrandeConservation account at Abiquiu Reservoir.

The rule includes an additional assignment statement to set the value for the AbiquiuMRGCD time series slot in the RuleHasFired data object for the current timestep to 1.0 using the user-defined RuleHasFiredSlot function in the Priority Functions utility group.

### **A.18.7. SetAbiquiuAlbuquerqueAccount**

*Explanation:* This rule sets the supply for the release of Albuquerque San Juan-Chama Project water for their surface water diversion.

*Rule Execution:* If the value in the AbiquiuAlbuquerque time series slot in the RuleHasFired data object for the current timestep is a NaN as determined with the user-defined RuleHasNOTFired function in the Priority Functions utility group and the result from the user-defined RuleIsInPriority function in the Priority Functions utility group (Refer to Section A.12.1.1) is TRUE, the rule fires.

*Rule Logic:* The value for the supply with a release type of Albuquerque and destination of Albuquerque\_Albuquerque as identified with the user-defined SuppliesByReleaseTypeDestination function in the Account Lists utility group is set to the minimum of the following three computed values: 1) the result from the user-defined AbiquiuAlbuquerqueDemand function in the Abiquiu Account Functions utility group, 2) the water in storage at Abiquiu Reservoir for the Albuquerque account at the previous timestep as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group plus the sum of the supplies at El Vado Reservoir with a destination of Albuquerque\_Abiquiu as identified with the user-defined AllSuppliesByReleaseTypeDestination function in the Account Lists utility group and summed using the user-defined SumSuppliesAtDate function in the Account Summing Functions utility and including consideration for the loss of the San Juan water from El Vado Dam to Abiquiu Reservoir as referenced using the user-defined SJCLoss function in the Get Data Functions utility group minus the sum of the supplies at Abiquiu Reservoir with a release type of AlbuquerqueLoan as identified with the user-defined AllSuppliesIntraByReleaseTypeDestination function in the Account Lists utility group and summed using the user-defined SumSuppliesAtDate function, or 3) the result from user-defined ReconcileSJOutflow function in the General Account Functions utility group minus the sum of all supplies for San Juan water at Abiquiu Reservoir minus the sum of the supplies from the RioGrandeConservation account at Abiquiu Reservoir.

The rule includes an additional assignment statement to set the value for the AbiquiuAlbuquerque time series slot in the RuleHasFired data object for the current timestep to 1.0 using the user-defined RuleHasFiredSlot function in the Priority Functions utility group.



### A.18.8. SetAbiquiuReclamationAccounts

*Explanation:* This rule sets the supply for the release from Reclamation San Juan-Chama Project water from Abiquiu Reservoir as needed to meet target flows in the Middle Valley.

*Rule Execution:* If the value in the AbiquiuReclamation time series slot in the RuleHasFired data object for the current timestep is a NaN as determined with the user-defined RuleHasNOTFired function in the Priority Functions utility group and the result from the user-defined RuleIsInPriority function in the Priority Functions utility group (Refer to Section A.12.1.1) is TRUE, the rule fires.

*Rule Logic:* The user-defined GetSortedMinFlowAccounts function in the Account Lists utility group is used to create a list of supplies and corresponding values to be set. The values are then set in the rule using a FOREACH loop. Within the function, a WITH DO loop is used with the user-defined PrioritizedAccounts function in the Priority Functions utility group to create a list of accounts with the corresponding priority as input to the AccountReleasePriority periodic slot in the AbiquiuData data object. The predefined SortPairsAscending function is used to sort the list based on the priorities. A second WITH DO loop is used to create a list of the Supplies to be set for all the accounts in that list which have a release type of Reclamation and destination of RioGrande\_RioGrande. An array is then created with the name for each supply and the corresponding value for the supply. If water is available to set the Supply, the value is set to the minimum of the amount of water in storage at Abiquiu Reservoir for the account as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group minus the sum of the supplies from that source as determined with the user-defined SumSupplies function in the Account Summing Function utility group plus the sum of the supplies into that source as determined with the user-defined SumAllSuppliesIntoSource function in the Account Lists utility group or the amount of San Juan water remaining to be allocated as tallied within the function.

The available water is computed in the rule as the minimum of either the value in the MinFlowsDemand time series slot in the AbiquiuData data object for the current timestep or the result from user-defined ReconcileSJOutflow function in the General Account Functions utility group minus the sum of all supplies for San Juan water at Abiquiu Reservoir plus the sum of the supplies from the RioGrandeConservation account at Abiquiu Reservoir.

This rule includes an additional assignment statement to set the value for the AbiquiuReclamation time series slot in the RuleHasFired data object for the current timestep to 1.0 using the user-defined RuleHasFiredSlot function in the Priority Functions utility group.

### **A.18.9. SetAbiquiuNMISCAccounts**

*Explanation:* This rule sets supply at Abiquiu with a release type of NMISC and destination of Jemez\_Jemez to payback debts related to the sediment pool at Jemez. The logic for this rule is the same as the logic for SetAbiquiuOtowipaybacks Rule except for the release type and destination. Refer to the discussion for SetAbiquiuOtowipaybacks Rule for more information on the logic in this rule.

### **A.18.10. AlbuquerqueToRioGrandeTransfer**

*Explanation:* If San Juan-Chama Project water was not released for the Albuquerque diversion, Albuquerque water is transferred to the Rio Grande account to make up for any shortage between the Albuquerque demand and the release from the Albuquerque account. Note that the demand includes a preemptive cutoff for when Abiquiu is operating at channel capacity and San Juan-Chama Project water would not be released.

*Rule Execution:* If the value for the supply to transfer water from the Albuquerque account at Abiquiu Reservoir to the Middle Valley as identified with the user-defined SupplyByReleaseTypeDestination function in the Account Lists utility group is NOT a NaN, the rule fires.

*Rule Logic:* The value for the supply at Lake Abiquiu with a release type of RioGrandePayback and destination of RioGrande\_Abiquiu as identified with the user-defined AllSuppliesIntraByReleaseTypeDestination function in the Account Lists utility group for the current timestep is set with this rule. An IF THEN ELSE statement is used to see if the value for the supply to transfer water from the Albuquerque account at Abiquiu Reservoir to the Middle Valley as identified with the user-defined SupplyByReleaseTypeDestination function in the Account Lists utility group is less than the result from the user-defined AbiquiuAlbuquerqueDemand function in the Abiquiu Account Functions utility group. If not, the value for the supply is set to 0.0 cfs. If so, the supply is set to the result from the AbiquiuAlbuquerqueDemand function minus the value for the supply to transfer water from the Albuquerque account at Abiquiu Reservoir to the Middle Valley. The AbiquiuAlbuquerqueDemand function computes the demand as 0.0 cfs if the Albuquerque diversion should be curtailed or it is before the input start year. Otherwise, the result is the minimum of the value in the Albuquerque time series slot in the MiddleValleyDemands data object at the next timestep or the amount of water in storage at Abiquiu Reservoir for the Albuquerque account as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group plus the sum of the supplies into that source as determined with the user-defined SumAllSuppliesIntoSource function in the Account Lists utility group.

## **A.19. Preparatory Cochiti And Jemez**

The rules in this policy group are used to determine whether storage at Cochiti Lake should be locked in as carryover storage per flood control operations and to compute the Rio Grande and San Juan releases from Cochiti Dam and Jemez Dam.

### **A.19.1. CochitiLockedIn**

*Explanation:* This rule is used to set a trigger indicating that storage should be locked in at Cochiti Lake as carryover storage until after the irrigation season per flood control operations.

*Rule Execution:* There are no execution constraints for this rule.

*Rule Logic:* This rule includes two assignment statements. The functions referenced for these assignments are the same as those used for locked in storage at Abiquiu Reservoir. Refer to the discussion for the AbiquiuLockedIn rule for more information on the logic used in this rule.

### **A.19.2. ComputeCochitiRGRelease**

*Explanation:* This rule computes the preliminary value for the release of Rio Grande water at Cochiti Dam. The rule includes a separate calculation if conservation storage at Cochiti Lake is being modeled (i.e. Cochiti deviations) where inflows of native Rio Grande water will be stored as not needed to meet downstream demands and then released as needed to meet target flows (to provide recruitment or overbank flows). Otherwise, the release is set to bypass inflows with any needed release from storage or adjustment from any potential storage during flood control operations and any potential non-irrigation season release of carrover storage from the irrigation season.

*Rule Execution:* If the value for the RGOutflow time series slot in the CochitiData data object for the current timestep is a NaN, the rule fires.

*Rule Logic:* the value for the RGOutflow time series slot in the CochitiData data object for the current timestep is set to the result from the user-defined CurrentRGInflow function in the General Compute Functions utility group plus the value for the Gain Loss for the Rio Grande account for the previous timestep as determined with the user-defined PreviousAccountGainLoss function in the Get Data Functions utility group plus the result from the user-defined CochitiRGStorageAdjustment function in the Cochiti Account Functions utility group plus the result from the user-defined RGCarryOverRelease function in the Flood Carry Over Functions utility group. The result is restricted using the predefined Max function to assure it is at least equal to the input value in the MinRGOutflow periodic slot in the CochitiData data object as referenced with the user-defined MinRGOutflow function in the Get Data Functions utility group. The result is

also checked against the physical restrictions of the outlet works using the user-defined `CheckThisResPhysicalConstraints` function (Refer to Section A.11.2.1).

The `CurrentRGInflow` function computes the Rio Grande inflow to Cochiti Lake as the value for the Inflow time series slot in the Cochiti storage reservoir object for the current timestep minus the sum of all the supplies at Abiquiu Reservoir that have a destination at or below Cochiti Dam. These supplies are identified with user-defined `AllSuppliesByReleaseTypeDestinations` function in the Account Lists utility group and summed using the user-defined `SumSuppliesAtDate` function in the Account Summing Functions utility group. The loss of San Juan water between Abiquiu Dam and Cochiti Lake is considered as referenced using the user-defined `SJCLoss` function in the Get Data Functions utility group.

Within the `RGCarryOverRelease` function, if the criteria for a carry over release are NOT satisfied, the value is set to 0.0 cfs. Otherwise, the value in the `RGCarryOverRelease` time series slot in the `CochitiData` data object for the current timestep is referenced. If that value is a NaN, the function then references the user-defined `ConstantRGCarryOverRelease` function in the Flood Carry Over Functions utility group.

### **A.19.3. ComputeCochitiRGConservationRelease**

*Explanation:* This rule computes the release of native Rio Grande water in conservation storage from Cochiti deviations. Releases are set to provide recruitment or overbank flows depending which operation is targeted for deviation operations based on set criteria. Releases are set to provide a peak release at the same time as the peak inflow. Releases may also be set to assure water is evacuated from conservation storage by the end of an input deviations period.

*Rule Execution:* If the value for the `RGConservationRelease` time series slot for the current timestep is a NaN, the rule fires.

*Rule Logic:* The release from conservation storage at Cochiti Lake is computed as the minimum flow needed for targets in the Middle Valley (including any higher Central targets set to provide recruitment or overbank flows) identified with the user-defined `MinimumFlowResult` function minus any release of San Juan-Chama Project water identified with the user-defined `CochitiSJOutflow` function and minus the value for the `RGOutflow` time series slot in the `CochitiData` data object for the current timestep. The needed release is also checked against the available supply as the previous storage in the `RioGrandeConservation` account at Cochiti checked using the user-defined `PreviousAccountStorage` function plus any inflow to conservation storage at the current timestep determined with reference to the user-defined `ComputeRGConsInflow` function minus any gain/loss on the `RioGrandeConservation` account. The release is also restricted to not exceed an input maximum. The timestep is checked to see if it is within a set period (30 days) after the beginning of conservation storage for deviations and before the end of an input deviations period (45 days) since the beginning on

conservation storage. The date to begin conservation storage could begin on an input date referenced with the GetStartCochitiDeviationsStorageDate, if a value is input, or would be computed as an offset from an estimated date for the peak inflow to Cochiti Lake computed with the user-defined CochitiMaxInflowDate function.

The release may be set higher with the user-defined ReleaseToEvacuateDeviationsRGConservationStorage function if needed to assure all conservation storage is evacuated by the end of the deviations period input to the DurationCochitiDeviationsPeriod scalar slot in the MiddleValleyDemands data object. Refer to Figure A.48 for a flowchart that shows the logic for setting releases from conservation storage at Cochiti Lake.

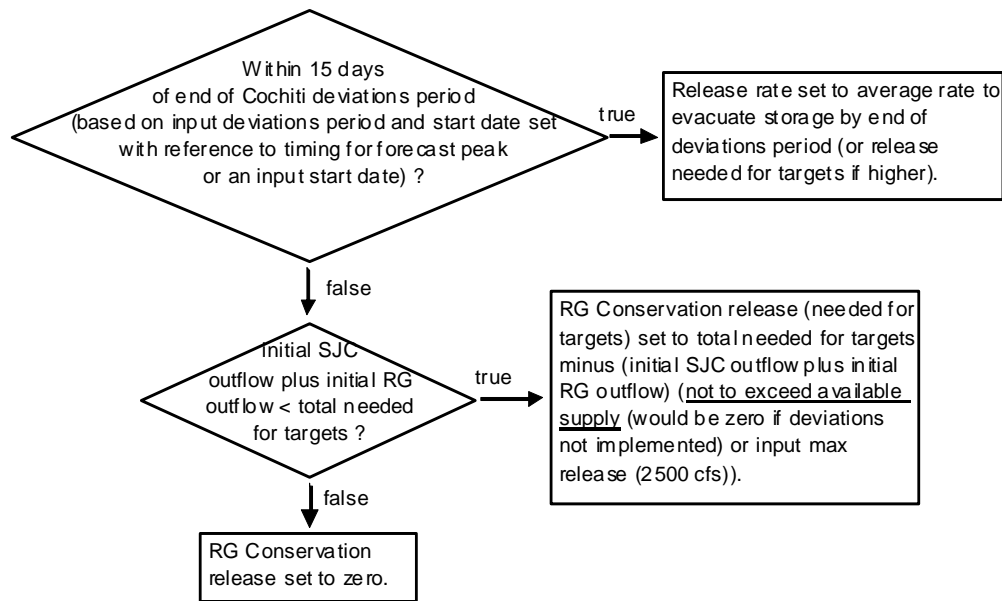


Figure A.48. Flowchart of Calculation of Release from Reregulation Storage per Cochiti Deviations

#### A.19.4. ComputeCochitiSJRelease

*Explanation:* This rule computes an initial value for the release of San Juan-Chama Project water from Cochiti Dam which is essentially a passthrough of all San Juan-Chama Project water except for Cochiti rec pool water..

*Rule Execution:* If the value for the RGOutflow time series slot in the CochitiData data object for the current timestep is NOT a NaN, the rule fires.

*Rule Logic:* The value for the SJOutflow time series slot in the CochitiData data object for the current timestep is set to the result from the user-defined CochitiSJOutflow function in the Cochiti Account Functions utility group. This function computes the release of San Juan water as the sum of all the supplies at Abiquiu Dam that have a

destination below Cochiti Dam (i.e. not including Cochiti rec pool water). These supplies are identified with user-defined AllSuppliesByReleaseTypeDestinations function in the Account Lists utility group and summed using the user-defined SumSuppliesAtDate function in the Account Summing Functions utility group. The loss rate for San Juan water between Abiquiu Dam and Cochiti Lake is considered as input to the Losses table slot in the SanJuanChamaRules data object and referenced using the user-defined SJCLoss function in the Get Data Functions utility group. Refer to Figure A.49 for a flowchart illustrating the logic used in this rule.

### **A.19.5. ComputeJemezRGRelease**

*Explanation:* This rule computes an initial value for the release of Rio Grande water at Jemez Dam as the sum of the inflow to the reservoir plus the amount of water in the Rio Grande storage account at Jemez Reservoir. If the amount of water in the storage account is negative, the outflow is set to the inflow multiplied by an input percentage.

*Rule Execution:* If the value for the RGOutflow time series slot in the JemezData data object for the current timestep is a NaN, the rule fires.

*Rule Logic:* The value for the RGOutflow time series slot in the JemezData data object for the current timestep is set to the result from the user-defined JemezRGOutflow function in the JemezFunctions utility group. This function includes a WITH DO loop to identify the value for the Inflow time series slot in the Jemez storage reservoir object for the current timestep using the user-defined CurrentInflow function in the Get Data Functions utility group (herein after referred to as the inflow). A second WITH DO loop is used to identify the value for the Storage in the Rio Grande account at Jemez Reservoir at the previous timestep using the user-defined PreviousAccountStorage function in the General Account Functions utility group (herein after referred to as the previous account storage). An IF THEN ELSE statement is used to see if the value for the inflow is less than or equal to zero and if the value for the previous account storage is less than or equal to zero. If so, the function result is set to 0.0 cfs. Otherwise, a second IF THEN ELSE statement is used to see if the previous account storage plus the volume of the inflow is less than zero but the inflow is greater than zero. If so, the result is set to the value for the inflow multiplied by the value in the PercentRGReleaseWhenNegRGStorage table slot in the JemezData data object. Otherwise, the result is set to the inflow plus the previous account storage. Refer to Figure A.49 for a flowchart illustrating the logic used in this rule.

### **A.19.6. SetJemezSJRelease**

*Explanation:* The San Juan release from Jemez Dam is set to zero.

*Rule Execution:* If the value for the SJOutflow time series slot in the JemezData data object for the current timestep is a NaN, the rule fires.

**Rule Logic:** The value for the SJOutflow time series slot in the JemezData data object for the current timestep is set to 0.0 cfs. A flowchart for this rule is presented in Figure A.49.

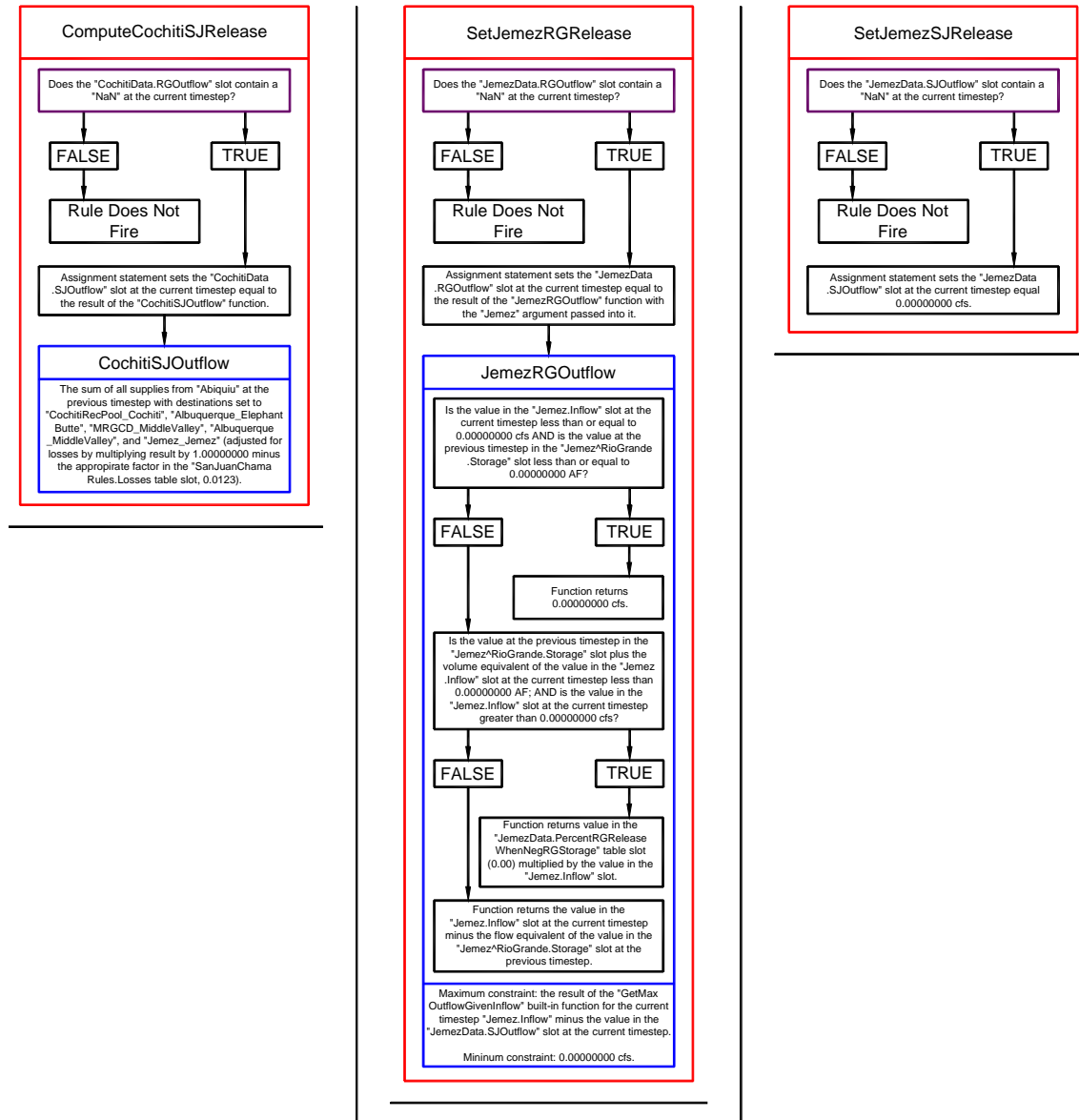


Figure A.49. Flowcharts for the ComputeCochitiSJRelease, SetJemezRGRelease, and SetJemezSJRelease Rules

## **A.20. Cochiti And Jemez**

The rules in this policy group are used to determine the total outflow from Cochiti Dam and Jemez Dam with consideration for downstream target flows, downstream channel capacities, stepped release restrictions, and flood control operations. Policy for balanced operations between Cochiti Dam and Jemez Dam for flood control operations is checked.

### **A.20.1. Cochiti Outflow**

*Explanation:* Unless a total outflow for Cochiti Dam has been input for the current timestep, the value is set to the maximum of either the total outflow to maintain the storage in the Cochiti Rec Pool account or the sum of the predetermined San Juan release with the Rio Grande release. The resulting total release is checked to see if it is physically legitimate based on the outlet works, and if not, the release is reset to reflect the restriction for the given outlet works.

*Rule Execution:* If the value for the Outflow time series slot in the Cochiti storage reservoir object for the current timestep is a NaN, the rule fires.

*Rule Logic:* This rule contains an IF THEN ELSE statement to see if the value in the TotalOutflow time series slot in the CochitiData data object for the current timestep as referenced using the user-defined TotalOutflow function in the Get Data Functions utility group is a NaN. If not, the value for the Outflow time series slot in the Cochiti storage reservoir object for the current timestep is set to that input TotalOutflow value. Otherwise, the value for the Outflow slot is set to the result from the user-defined InitialCochitiOutflow function in the Cochiti Account Functions utility group.

The InitialCochitiOutflow function includes another check to see if the TotalOutflow slot is a NaN. A second IF THEN ELSE statement is then used to see if the value for the Outflow time series slot in the Abiquiu level power reservoir object for the previous timestep is greater than or equal to the sum of the values in the MRGCDDemand and MinFlowsDemand time series slots in the AbiquiuData data object for the current timestep and if the value for the Inflow time series slot in the Cochiti storage reservoir object for the current timestep is less than the value for the MinLockinFlow table slot in the FloodCarryOverData data object and if it is the irrigation season (March 1 through October 31) as determined with the user-defined IrrigationSeason function in the Date Functions utility group. If those criteria are satisfied, the total outflow is set to the maximum of either the result from the predefined SolveOutflow function as used to compute the outflow to yield a storage at Cochiti Lake equal to the Storage for the CochitiRecPool account plus the result from the user-defined CochitiRGStorageAdjustment function in the Cochiti Account Functions utility group or the sum of the values for the SJOutflow and RGOutflow time series slots in the CochitiData data object for the current timestep as referenced with the user-defined SJOutflow and RGOutflow functions in the General Account Function utility group. If



those criteria are not satisfied, a third IF THEN ELSE statement is used to see if the value for the Outflow time series slot in the Abiquiu level power reservoir object for the previous timestep minus the sum of the supplies at Abiquiu Reservoir with a destination of the CochitiRecPool\_Cochiti as identified with user-defined

AllSuppliesByReleaseTypeDestinations function in the Account Lists utility group and summed using the user-defined SumSuppliesAtDate function in the Account Summing Functions utility group is equal to the value in the MRGCDDemand time series slot in the AbiquiuData data object for the current timestep and if it is the irrigation season as determined with the user-defined Irrigation Season function. If so, the outflow is set to the value in the MRGCD time series slot in the MiddleValleyDemands data object. Otherwise, the same equation used under the second IF THEN ELSE statement is referenced again.

Within the rule, the result for the Outflow is checked against the physical restrictions of the outlet works using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1).

## **A.20.2. JemezOutflow**

*Explanation:* Unless a total outflow for Jemez Dam has been input for the current timestep, the value is set to the sum of the predetermined Rio Grande and San Juan release unless the reservoir needs to capture flow for the exchange of San Juan water for the sediment pool. Then, the outflow is set to an input release rate as that exchange is taking place. The resulting total release is checked to see if it is physically legitimate based on the outlet works, and if not, the release is reset to reflect the restriction for the given outlet works.

*Rule Execution:* If the value for the Outflow time series slot in the Jemez storage reservoir object for the current timestep is a NaN, the rule fires.

*Rule Logic:* This rule contains an exterior IF THEN ELSE statement to see if the value in the TotalOutflow time series slot in the JemezData data object for the current timestep as referenced with the user-defined TotalOutflow function in the Get Data Functions utility group is a NaN. If not, the value for the Outflow time series slot in the Jemez storage reservoir object for the current timestep is set to that input TotalOutflow value.

Otherwise, a second IF THEN ELSE statement is used to see if it is not the start timestep as determined with the user-defined IsNotFirstTimestep function in the General Boolean Functions utility group and if the sum of the values in the NMISCJemez time series slot in the DeliveryRequests data object year-to-date as determined with the user-defined JemezAccumulatedDeliveryRequestInACRE-FT function minus the sum of the supplies for the RioGrandeJemezToRioGrandeJemezJemezSedimentPoolJemez transfer year-to-date as determined with the user-defined JemezAccumulatedSJCaptureInACRE-FT function in the JemezFunctions utility group is greater than 1.0 acre-ft. If not, the outflow is set to the sum of the values for the SJOutflow and RGOutflow time series slots in the JemezData data object for the current timestep as referenced with the user-defined

SJOutflow and RGOutflow functions in the General Account Function utility group. If so, the outflow is set to the result from EstimateJemezOutflowForExchange function in the JemezFunctions utility group. Within that function, an exterior IF THEN ELSE statement is used to see if it is NOT the irrigation season (March 1 through October 31) as determined with the user-defined IrrigationSeason function in the Date Functions utility group. If that criterion is satisfied, the outflow is set to 0.0 cfs. Otherwise, a second IF THEN ELSE statement is used to see if the estimated inflow to Jemez Reservoir as determined with the user-defined EstimateJemezInflow function in the JemezFunctions utility group is greater than the value for the MinNonWinterEXRelease in the ReservoirData table slot in the JemezData data object or if the value for the Storage in the Rio Grande account at Jemez Reservoir as referenced with the user-defined PreviousRGStorage function in the General Account Functions utility group is greater than the volume of the MinNonWinterEXRelease. If so, the outflow is set to the value for the MinNonWinterEXRelease. Otherwise, the outflow is set to 0.0 cfs.

Within the rule, the result for the Outflow is checked against the physical restrictions of the outlet works using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1).

### **A.20.3. CentralChannelCapacityRule**

*Explanation:* This rule uses hypothetical simulation to determine the release from Cochiti Dam to meet the input channel capacity at Central.

*Rule Execution:* If the value for the MaxReleaseForCentralChannelCap time series slot in the CochitiData data object for the current timestep is a NaN and the value for the EstimatedInflow time series slot in the CochitiData data object for the current timestep is greater than 2500 cfs, the rule fires

*Rule Logic:* The value for the MaxReleaseForCentralChannelCap time series slot in the CochitiData data object for the current timestep is set to the result from the user-defined CochitiReleaseForCentralChannelCapacity function in the Cochiti Channel Capacity Functions utility group. This function uses the predefined HypTargetSimWithStatus function to determine the required outflow from Cochiti Dam (or the value for the Inflow time series slot for the BlwCochitiDiversionsReach reach object) to yield the channel capacity at Central (or the Gage Inflow time series slot in the Central stream gage object) at the next timestep equal to the input value in the CentralChannelCapacity periodic slot in the CochitiData data object. The function completes the hypothetical simulation for all the objects in the BlwCochitiToCentral subbasin and considers a check on the minimum outflow and maximum outflow from Cochiti Dam as input to the MinRelease periodic slot and as the maxRelease in the ReservoirData table slot in the CochitiData data object and referenced using the user-defined MinOutflow and MaxOutflow functions in the Get Data Functions utility group. The tolerance is input to the CentralChannelCapacity periodic slot in the CochitiData data object. The release from Jemez Dam (or the value for the Gage Inflow time series slot in the BlwJemez stream gage object) is assumed to be

zero for the next timestep for the hypothetical simulation as identified with the user-defined EstimatedJemezReleaseList function in the HypotheticalSimulation Functions utility group.

#### **A.20.4. SanMarcialChannelCapacityRule**

*Explanation:* This rule uses hypothetical simulation to determine the release from Cochiti Dam to meet the input channel capacity at San Marcial.

*Rule Execution:* If the value for the MaxReleaseForSanMarcialChannelCap time series slot in the CochitiData data object for the current timestep is a NaN and the value for the EstimatedInflow time series slot in the CochitiData data object for the current timestep is greater than 2500 cfs, the rule fires.

*Rule Logic:* The value for the MaxReleaseForSanMarcialChannelCap time series slot in the CochitiData data object for the current timestep is set to the result from the user-defined CochitiReleaseForSanMarcialChannelCapacity function in the Cochiti Channel Capacity Functions utility group. This function uses the predefined HypTargetSimWithStatus function to determine the required outflow from Cochiti Dam (or the value for the Inflow time series slot for the BlwCochitiDiversionsReach reach object) to yield the channel capacity at San Marcial (or the Gage Inflow time series slot in the SanMarcialFloodway stream gage object) at the timestep after the next timestep equal to the input value in the SanMarcialChannelCapacity periodic slot in the CochitiData data object. The function completes the hypothetical simulation for all the objects in the BlwCochitiToSanMarcial subbasin and considers a check on the minimum outflow and maximum outflow from Cochiti Dam as input to the MinRelease periodic slot and as the maxRelease in the ReservoirData table slot in the CochitiData data object and referenced using the user-defined MinOutflow and MaxOutflow functions in the Get Data Functions utility group. The tolerance is input to the SanMarcialChannelCapacity periodic slot in the CochitiData data object. The release from Jemez Dam (or the value for the Gage Inflow time series slot in the BlwJemez stream gage object) is assumed to be zero for the next timestep and the timestep after the next timestep for the hypothetical simulation as identified with the user-defined EstimatedJemezReleaseList function in the HypotheticalSimulation Functions utility group.

#### **A.20.5. JemezSanMarcialChannelCapacity**

*Explanation:* This rule simply sets the value for the MaxReleaseForSanMarcialChannelCap time series slot in the JemezData data object for the current timestep equal to the value in the MaxReleaseForSanMarcialChannelCap time series slot in the CochitiData data object for the current timestep.

## **A.20.6. CochitiChannelCapacityRestrictions**

*Explanation:* This rule resets the outflow from Cochiti Dam and Jemez Dam to comply with the downstream channel capacities at Central and San Marcial if necessary.

*Rule Execution:* The rule fires if the rule has not already fired for the current timestep as checked with reference to the predefined HasRuleFiredSuccessfully function and if the sum of the values in the Outflow time series slots in the Cochiti and Jemez storage reservoir objects minus the minimum of the values for the MaxReleaseForCentralChannelCap and MaxReleaseForSanMarcialChannelCap time series slots in the CochitiData data object as determined with the user-defined CochitiReleaseForChannelCapacity function in the Cochiti Channel Capacity Functions utility group is greater than the tolerance for the channel capacity at the SanMarcial as input to the SanMarcialChannelCapacity periodic slot in the CochitiData data object and the value for the EstimatedInflow time series slot in the CochitiData data object for the current timestep is greater than 2500 cfs.

*Rule Logic:* If the execution constraints are satisfied, three assignments are made in this rule. The value for the Outflow time series slot in the Cochiti storage reservoir object is set to the minimum of the values for the MaxReleaseForCentralChannelCap and MaxReleaseForSanMarcialChannelCap time series slots in the CochitiData data object as determined with the user-defined CochitiReleaseForChannelCapacity function minus the result for the user-defined JemezFloodRelease function in the Jemez Flood Control Functions utility group.

For the second assignment, the value for the Outflow time series slot in the Jemez storage reservoir object is set to the result for the user-defined JemezFloodRelease function. This function sets the Jemez release equal to the result from the user-defined CochitiReleaseForChannelCapacity function multiplied by the storage space at Jemez Reservoir as determined with the user-defined AvailableStorage function in the Get Data Functions utility group divided by the total storage space at both Jemez Reservoir and Cochiti Lake. The predefined Min function is used to assure the release is no greater than the value in the Inflow time series slot at Jemez Reservoir plus the storage in the Rio Grande account at Jemez Reservoir at the previous timestep as determined with the user-defined PreviousAccountStorage function in the General Account Functions utility group and converted to a flow using the predefined VolumeToFlow function.

For both of the previous assignments, the result for the Outflow is checked against the physical restrictions of the outlet works for the corresponding dam using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1).

## **A.20.7. CochitiSteppedReleaseWithCochitiDeviation**

*Explanation:* The outflow from Cochiti Dam (and Jemez Dam) may be reset to assure increasing or decreasing releases are stepped if needed. This rule determines whether stepped releases are required, and if so, computes the release for Cochiti (and Jemez).

*Rule Execution:* The rule fires if the rule has not already fired for the current timestep as checked with reference to the predefined HasRuleFiredSuccessfully function.

*Rule Logic:* Three assignments are included in this rule. First, an IF THEN statement is used to see if the result from the user-defined SteppedReleaseIsNeeded function in the Stepped Release Functions utility group is satisfied for Cochiti Lake and if the Central target is increasing. The SteppedReleaseIsNeeded function includes three separate functions to check the change to the outflow as computed using the user-defined ComputeDeltaOutflow function in the General Compute Functions utility group against different criteria. If the function is satisfied, the value in the Outflow time series slot in the Cochiti storage reservoir object for the current timestep is reset to the result from the user-defined DetermineSteppedRelease function in the Stepped Release Functions utility group. This function includes an exterior IF THEN ELSE statement to see if the outflow from Cochiti Dam is increasing as determined using the user-defined OutflowIsIncreasing function in the General Boolean Functions utility group. If so, the outflow is calculated with the user-defined ComputeOutflowIncrSteppedRelease function. If not, the outflow is computed using the user-defined ComputeOutflowDecrSteppedRelease function in the Stepped Release Functions utility group unless it is determined that the release should be shutoff.

The same logic is used to determine if a stepped release from Jemez Dam is needed, and if so, the release is computed using the same functions. The results for both cases are checked against the physical restrictions of the outlet works for the corresponding dam using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1). Refer to Figures A.45 through A.47 for flowcharts that illustrate the logic used for setting stepped releases.

## **A.20.8. CochitiWCMBalancedRelease**

*Explanation:* This rule resets the outflow from Cochiti Dam and Jemez Dam as necessary to balance operations and the available flood storage space at each reservoir as stipulated in the water control manuals for each dam.

*Rule Execution:* This rule fires if the rule has not already fired for the current timestep as checked with reference to the predefined HasRuleFiredSuccessfully function and if the value in the Outflow time series slots in the Cochiti storage reservoir object minus the result from the user-defined CochitiBalancedOperation function in the Cochiti Jemez WCM Balance Functions utility group is greater than 10 cfs or if the value in the Outflow time series slots in the Jemez storage reservoir object minus the result from the user-defined JemezBalancedOperation function in the Cochiti Jemez WCM Balance Functions

utility group is greater than 10 cfs and if the result from the user-defined IfCochitiJemezBalancedOperation function is TRUE.

*Rule Logic:* If the execution constraints are satisfied, three assignments are made in this rule. The value for the Outflow time series slot in the Cochiti storage reservoir object is set to the result from the user-defined CochitiBalancedOperation function. Within this function, an IF THEN ELSE statement is used to see if the result from the user-defined CochitiJemezStorageDifferentialRatio function in the Cochiti Jemez WCM Balanced Functions utility group is less than zero. The result from this function is the ratio of the currently available flood storage space to the total available flood storage space at Cochiti Lake minus the ratio of the currently available flood storage space to the total available flood storage space at Jemez Reservoir. So, if that value is less than zero (the ratio at Cochiti Lake is less than the ratio at Jemez Reservoir), the outflow is increased using that ratio. Otherwise, the outflow is decreased based on the adjustment made to the release from Jemez Dam. For the second assignment, the user-defined JemezBalanceOperation function is used to set the value for the Outflow time series slot in the Jemez storage reservoir object. The same ratio is checked again, and the outflow is adjusted similarly.

For both of the previous assignments, the result for the Outflow is checked against the physical restrictions of the outlet works for the corresponding dam using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1). Refer to Figures A.50 through A.52 for flowcharts that depict the logic used for balanced operations.

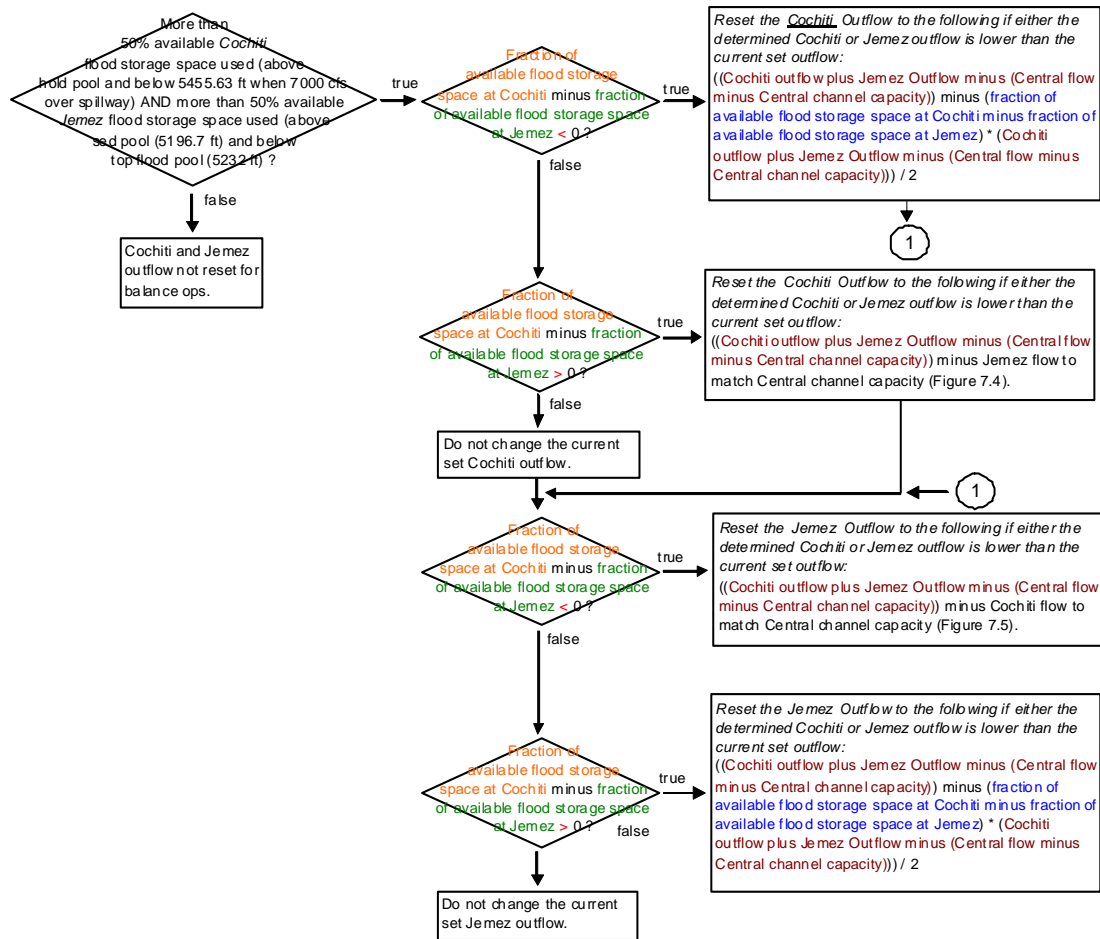


Figure A.50. Flow Chart with Logic for Setting Cochiti and Jemez Balanced Operations

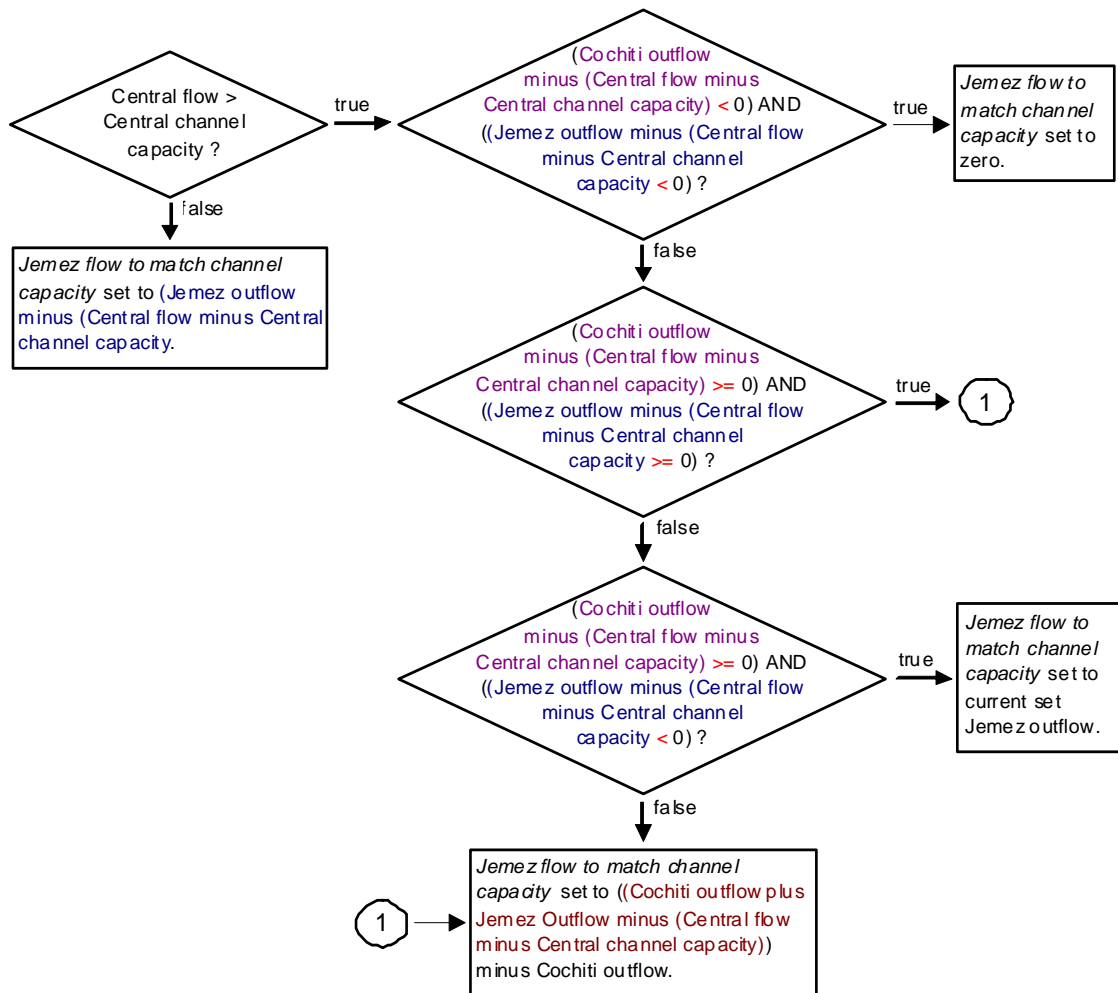


Figure A.51. Flow Chart for Setting Jemez Flow for Channel Capacity – Balanced Ops



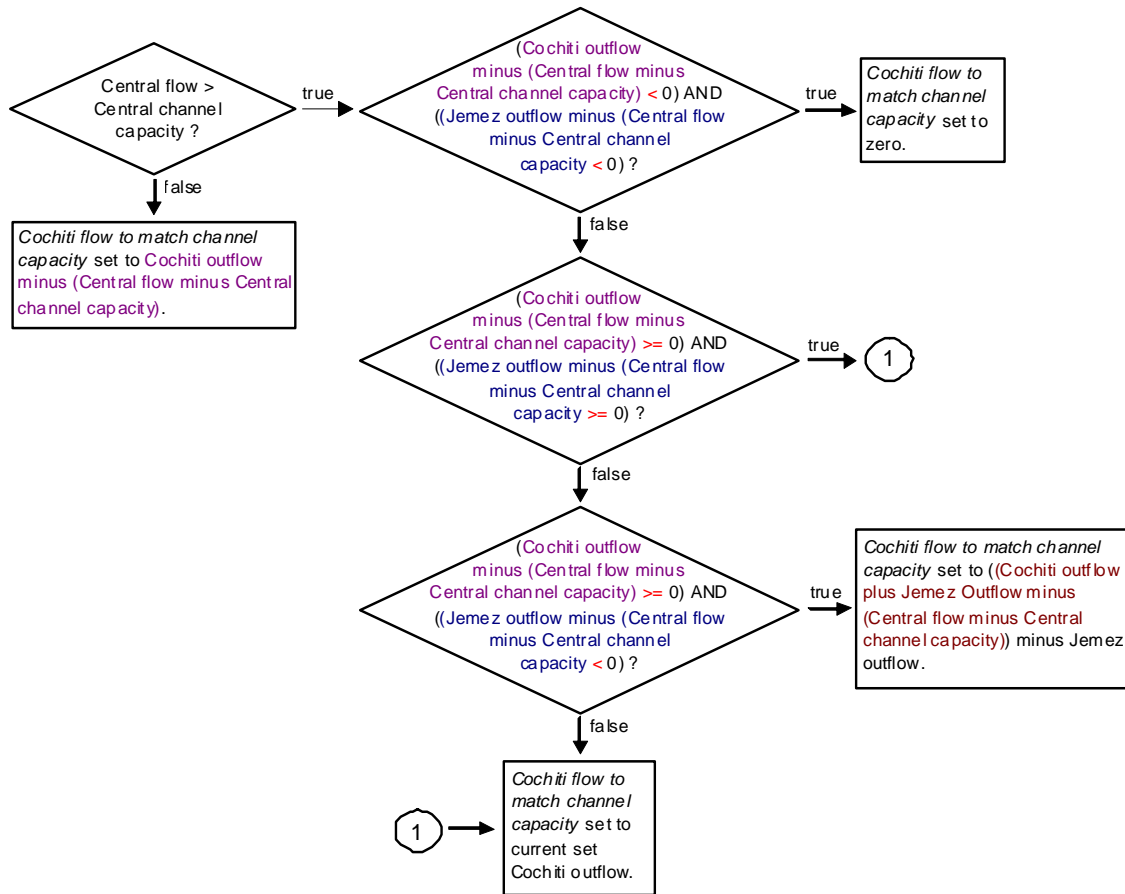


Figure A.52. Flow Chart for Setting Cochiti Flow for Channel Capacity – Balanced Ops

### A.20.9. CochitiFloodControl

*Explanation:* If there is an unregulated spill, per flood control operations, this rule resets the outflow from Cochiti Dam to match the channel capacity at Central if necessary and as possible.

*Rule Execution:* This rule fires if the rule has not already fired for the current timestep as checked with reference to the predefined HasRuleFiredSuccessfully function and the value for the Unregulated Spill time series slot in the Cochiti storage reservoir object is greater than zero as determined with the user-defined ReservoirIsSpillingUnreg function in the General Boolean Functions utility group.

*Rule Logic:* If the execution constraints are satisfied, the value for the Outflow time series slot in the Cochiti storage reservoir object for the current timestep is reset to the result from the CochitiFCOutflow function in the Cochiti Flood Control Functions utility group. This function includes an IF THEN ELSE statement to see if the value for the Unregulated Spill time series slot in the Cochiti storage reservoir object for the current timestep is less than the channel capacity at Cochiti as input in the ChannelCapacities

table slot in the CochitiData data object and referenced with the user-defined ChannelCapacity function in the General Channel Capacity Functions utility group. If not, the function result is set equal to the value in the Outflow time series slot in the Cochiti storage reservoir object for the current timestep (i.e. the outflow is not changed). If so, the function result is then computed using the user-defined CochitiFlowToMatchCentralChannelCapacity function in the Cochiti Channel Capacity Functions utility group. This function computes the outflow to match the input channel capacity, if necessary, based on the value in the Gage Inflow time series slot in the Central stream gage object for the current timestep. An exterior IF THEN ELSE statement is used to see if the value in the Gage Inflow time series slot is greater than the channel capacity. If not, the outflow at the current timestep is adjusted based on that difference to match the channel capacity. If so, a second IF THEN ELSE statement is used to see if the computed release to match the channel capacity is less than zero as identified with the user-defined IsReleaseToMatchChannelCapacity<0 function in the General Channel Capacity Functions utility group. If so, the release is set to the value input to the Minimum column in the ChannelCapacities table slot in the CochitiData data object as referenced using the user-defined ChannelCapacity function. Otherwise, a third IF THEN ELSE statement is used to see if the computed release to match the channel capacity is greater than zero as identified with the IsReleaseToMatchChannelCapacity>0 function. If so, the outflow is set to the Outflow time series slot in the Cochiti storage reservoir object for the current timestep (i.e. the outflow is not changed). The last IF THEN ELSE statement identifies whether the outflow from just Jemez Dam to match the target is less than zero, and then the outflow from Cochiti Dam is adjusted accordingly.

Within the rule, the result is checked against the physical restrictions of the outlet works using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1).

## **A.20.10. CochitiRGCarryOver**

*Explanation:* This rule sets a value for the release of carry over storage and sets the remaining carry over content. Flood water may be carried over until after the irrigation season. This rule computes the subsequent release and remaining carry over storage.

*Rule Execution:* The rule fires if the value for the RGCarryOverRelease time series slot in the CochitiData data object for the current timestep is a NaN.

*Rule Logic:* The value for the RGCarryOverRelease time series slot in the CochitiData data object for the current timestep is set to the result from the user-defined RGCarryOverRelease function in the Flood Carry Over Functions utility group. This function includes an exterior IF THEN ELSE statement to see if it is the carry over release season (November through March) as determined using the user-defined FloodCarryOverReleaseSeason function in the Date Functions utility group and if the value for the Storage in the Rio Grande account is greater than zero and if the value for the Locked In time series slot in the Cochiti storage reservoir object is equal to zero. If

not, the carry over release is set to 0.0 cfs. If so, a second IF THEN ELSE statement is used to see if the result from the user-defined IfRGCarryOverRelease function in the Flood Carry Over Functions utility group is true. This function checks to see if the value for the RGCarryOverLeft time series slot in the CochitiData data object is greater than zero or the value for the Storage in the Rio Grande account at Cochiti Lake is greater than the value in the MinRGCarryOverStorage table slot in the CochitiData data object or the value in the Carryover Content time series slot in the Cochiti storage reservoir object for October 31 of the current year is greater than zero. If not, the carry over release is set to 0.0 cfs. If so, the carry over release is set to the value in the RGCarryOverRelease time series slot in the CochitiData data object for the current timestep unless it is a NaN. Then, the value is set to the result from the user-defined ConstantRGCarryOverRelease function in the Flood Carry Over Functions utility group.

The rule includes a second assignment statement to set the value for the RGCarryOverLeft time series slot in the CochitiData data object for the current timestep to the result from the user-defined RGCarryOverLeft function in the Flood Carry Over Functions utility group. Within this function, an exterior IF THEN ELSE statement is used to see if the results from the RGCarryOverIsNaNBoolean function and the user-defined ConstantCOREleaseBoolean function in the Flood Carry Over Boolean Functions utility group are true. The former checks to see if the result from the FloodCarryOverReleaseSeason function is true and if the Storage in the Rio Grande account at the previous timestep is greater than zero and if the result from the IfRGCarryOverRelease function is true and if the value for the RGCarryOverRelease time series slot in the CochitiData data object for the current timestep is a NaN. The latter checks some of the same criteria and also to see if the value for the Locked In time series slot in the Abiquiu level power reservoir object is equal to 0.0. If those criteria are satisfied, a second IF THEN ELSE statement checks again to see if the Storage in the Rio Grande account at the previous timestep is greater than zero. If not, the function result is 0.0 acre-ft. A third IF THEN ELSE statement is then used to see if the value for the RGCarryOverLeft time series slot in the CochitiData data object for the previous timestep is greater than zero. If not, the carry over left is set to the previous Storage in the Rio Grande account minus the volume of the RGCarryOverRelease. A fourth IF THEN ELSE statement is used to see if the result from the user-defined IsRGConservationAllowed function in the RGConservationFunctions utility group is true. If not, the carry over left is the value in the RGCarryOverLeft time series slot in the CochitiData data object for the previous timestep minus the RGCarryOverRelease. If so, the result is the value for the RGCarryOverLeft minus the result from the user-defined ComputeAbiquiuRGConsInflow function minus the RGCarryOverRelease. If the exterior IF THEN ELSE is NOT satisfied, a second IF THEN ELSE statement is used to see if the results from the RGCarryOverNOTNaNBoolean function and the ConstantCOREleaseBoolean functions are true. If so, the Storage in the Rio Grande account is checked again to see if it is greater than zero and then, the result is the previous Storage in the Rio Grande account minus the RGCarryOverRelease. If not, an IF THEN ELSE statement is used to see if the value for the RGCarryOverRelease is a NaN. If so, the result from the user-defined ComputeIsNaNRGCarryOverLeft function is checked to see if it is less than or equal to zero or the RGCarryOverRelease is equal to

zero. If so, the result is 0.0 acre-ft. If not, the function result is the result from the user-defined ComputeIsNaNRGCarryOverLeft function. Otherwise, a last IF THEN ELSE statement is used to see if the ComputeNOTIsNaNRGCarryOverLeft function is less than or equal to zero or the RGCarryOverRelease is greater than or equal to the Storage in the Rio Grande account. If so, the result is 0.0 acre-ft. If not, the carry over left is set to the result from the user-defined ComputeNOTIsNaNRGCarryOverLeft function.

## **A.21. Cochiti and Jemez Accounts**

The rules in this policy group are used to set the supplies to transfer Rio Grande and San Juan water downstream from Cochiti Dam and Jemez Dam and also transfer water to the sediment pool account at Jemez Reservoir. In addition, the Rio Grande and San Juan releases are reconciled.

### **A.21.1. SetCochitiRGConservationAccountInflow**

*Explanation:* This rule sets the accounting supply to transfer Rio Grande inflows to Cochiti Lake to conservation storage based on the defined space available for storage for deviations operations.

*Rule Execution:* If the value for the accounting supply to transfer water from the Rio Grande account at Cochiti Lake to the RioGrandeConservation account is a NaN, the rule fires.

*Rule Logic:* The value for the accounting supply to transfer water from the Rio Grande account at Cochiti Lake to the RioGrandeConservation account is set with an IF THEN ELSE statement. If the accrual to the account is less than the space available identified with the GetRGConservationSpaceAvailable function and the timestep is prior to 15 days before the determined end of the deviations period, the supply is set to the result from the user-defined ComputeRGConsInflow function not to exceed the maximum inflow referenced with the user-defined MaxRGConservationInflow function. Otherwise, the accounting supply is set to zero cfs.

### **A.21.2. SetCochitiRGAccount**

*Explanation:* This rule sets the supply for the release of Rio Grande water from Cochiti Dam as the difference in the final total outflow and the outflow of San Juan-Chama Project water.

*Rule Execution:* If the value for the accounting supply for the release from the Rio Grande account at Cochiti Lake as identified with the user-defined SuppliesByReleaseTypeDestination function in the Account Lists utility group is a NaN, the rule fires.

*Rule Logic:* The value for the accounting supply for the release from the Rio Grande account at Cochiti Lake as identified with the user-defined `SuppliesByReleaseTypeDestination` function in the Account Lists utility group is set using the user-defined `ReconcileRGOutflow` function in the General Account Functions utility group. For Cochiti Lake, the result is set equal to the value in the Outflow time series slot in the Cochiti storage reservoir object for the current timestep minus the value in the SJOoutflow time series slot in the CochitiData data object for the current timestep as identified with the user-defined `SJOoutflow` function in the General Account Functions utility group. The predefined `Max` function is used to restrict the result to be no less than the value in the `MinRGOutflow` periodic slot in the CochitiData data object as referenced with the user-defined `MinRGOutflow` function in the Get Data Functions utility group.

### **A.21.3. CheckCochitiRGOutflow**

*Explanation:* Aborts the simulation if the supply for the release of Rio Grande water from the Rio Grande account at Cochiti Lake was not set. This rule helps with debugging because a simulation will stop if the Rio Grande supply was not set and the problem and any missing input can be identified by backtracking in the calculations from the point in the simulation that the run aborted.

*Rule Execution:* There are no execution constraints for this rule.

*Rule Logic:* If the value for the Supply at Cochiti Lake with a release type of Rio Grande and destination of `RioGrande_RioGrande` is a NaN, the `STOP_RUN` command is used to stop the simulation, and the `PRINT` statement is used to print a comment to the Diagnostics Output window that the run stopped because the account slot value was not set.

### **A.21.4. SetCochitiFlowThroughAccounts**

*Explanation:* This rule sets the accounting supplies for flowthrough accounts at Cochiti Lake.

*Rule Execution:* If the value in the `CochitiFlowThrough` time series slot in the `RuleHasFired` data object for the current timestep is a NaN as determined with the user-defined `RuleHasNOTFired` function in the Priority Functions utility group and the result from the user-defined `RuleIsInPriority` function in the Priority Functions utility group (Refer to Section A.12.1.1) is `TRUE`, the rule fires.

*Rule Logic:* The user-defined `GetSortedFlowThruAccounts` function in the Account Lists utility group is used to create a list of supplies and corresponding values to be set. The values are then set in the rule using a `FOREACH` loop. Refer to the discussion of the rule logic for the `SetAbiquiuFlowThroughAccounts` for more details on this function. The

function is used in the same manner here for Cochiti Lake. This rule includes an additional assignment statement to set the value for the CochitiFlowThrough time series slot in the RuleHasFired data object for the current timestep to 1.0 using the user-defined RuleHasFiredSlot function in the Priority Functions utility group.

#### **A.21.5. SetCochitiReclamationAccounts**

*Explanation:* This rule sets the accounting supply for the release from conservation storage as needed for targets in the Middle Valley.

*Rule Execution:* If the value in the CochitiReclamation time series slot in the RuleHasFired data object for the current timestep is a NaN as determined with the user-defined RuleHasNOTFired function in the Priority Functions utility group and the result from the user-defined RuleIsInPriority function in the Priority Functions utility group (Refer to Section A.12.1.1) is TRUE, the rule fires.

*Rule Logic:* The user-defined GetSortedMinFlowAccounts function in the Account Lists utility group is used to create a list of supplies and corresponding values to be set. The values are then set in the rule using a FOREACH loop. Refer to the discussion of the rule logic for the SetAbiquiuReclamationAccounts for more details on this function. The function is used in the same manner here for Cochiti Lake for setting the release from conservation storage as needed to meet targets in the Middle Valley. This rule includes an additional assignment statement to set the value for the CochitiReclamation time series slot in the RuleHasFired data object for the current timestep to 1.0 using the user-defined RuleHasFiredSlot function in the Priority Functions utility group.

#### **A.21.6. RGTransfer**

*Explanation:* This rule sets the accounting supply for the transfer of Rio Grande water at Jemez Reservoir to the sediment pool account as part of the NMISC Jemez sediment pool exchange.

*Rule Execution:* There are no execution constraints for this rule.

*Rule Logic:* The value for the Supply for the RioGrandeJemezToRioGrandeJemezJemezSedimentPoolJemez transfer for the current timestep is set to the result from the user-defined JemezExchange function in the JemezFunctions utility group. This function includes an exterior IF THEN ELSE statement to see if it is not the start timestep as determined with the user-defined IsNotFirstTimestep function in the General Boolean Functions utility group and if the sum of the supplies for the RioGrandeJemezToRioGrandeJemezJemezSedimentPoolJemez transfer year-to-date as determined with the user-defined JemezAccumulatedSJCaptureInACRE-FT function is less than the sum of the values in the NMISCJemez time series slot in the

DeliveryRequests data object year-to-date as determined with the user-defined JemezAccumulatedDeliveryRequestInACRE-FT function in the JemezFunctions utility group and if the Storage in the Rio Grande account at Jemez Reservoir at the previous timestep as determined with the user-defined PreviousRGStorage function in the General Account Functions utility group is greater than the result from the JemezAccumulatedDeliveryRequestInACRE-FT function minus the result from the JemezAccumulatedSJCaptureInACRE-FT function and if the JemezAccumulatedDeliveryRequestInCFS function minus the result from the JemezAccumulatedSJCaptureInCFS function is greater than the value in the minExchangeRGFlow column in the ExchangeData table slot in the JemezData data object and if the result from the user-defined JemezIsStoring function in the JemezFunctions utility group is true which checks to see if the result from the user-defined EstimateJemezInflow function minus the result from the user-defined EstimateJemezOutflowForExchange function in the JemezFunctions utility group is greater than zero. If all those criteria are satisfied, the supply is set to the minimum of either the result from the user-defined EstimateJemezInflow function minus the result from the user-defined EstimateJemezOutflowForExchange function or the maximum of either the result from the user-defined ComputeJemezExchangeWhenRGStorage function in the JemezFunctions utility group or the value in the minExchangeRGFlow column in the ExchangeData table slot. If all the aforementioned criteria are NOT satisfied, a second IF THEN ELSE statement is used to see if it is not the start timestep as determined with the user-defined IsNotFirstTimestep function and if the result from the user-defined JemezAccumulatedSJCaptureInACRE-FT function is less than the result from the user-defined JemezAccumulatedDeliveryRequestInACRE-FT function and if the result from the user-defined JemezIsStoring function in the JemezFunctions utility group is true. If not, the supply is set to 0.0 cfs. If so, the supply is set to the minimum of either the result from the user-defined EstimateJemezInflow function minus the result from the user-defined EstimateJemezOutflowForExchange function or the maximum of either the result from JemezAccumulatedDeliveryRequestInCFS function minus the result from the JemezAccumulatedSJCaptureInCFS function or the value in the minExchangeRGFlow column in the ExchangeData table slot.

### **A.21.7. SetJemezAccounts**

*Explanation:* This rule sets the accounting supplies for the release of Rio Grande water at Jemez Dam and the release of any MRGCD San Juan-Chama Project water in Jemez Reservoir.

*Rule Execution:* If the value for the supply to transfer water from the Rio Grande account at Jemez Reservoir to the destination of RioGrande\_RioGrande as identified with the user-defined AllSuppliesByReleaseTypeDestination function in the Account Lists utility group is a NaN, the rule fires.

*Rule Logic:* There are two assignment statements in this rule. First, the value for the supply to transfer water from the Rio Grande account at Jemez Reservoir to the

destination of RioGrande\_RioGrande as identified with the user-defined AllSuppliesByReleaseTypeDestination function in the Account Lists utility group is set using the user-defined ReconcileRGOutflow function in the General Account Functions utility group. For Jemez Reservoir, this function includes an IF THEN ELSE statement to see if the value for the RGOutflow time series slot in the JemezData data object for the current timestep as referenced using the user-defined RGOutflow function in the General Account Functions utility group is greater than the value in the Outflow time series slot in the Jemez storage reservoir object. If so, the function result is set to that Outflow for the Jemez storage reservoir object. Otherwise, a second IF THEN ELSE statement is used to see if the value for that Outflow is greater than the sum of the values in the RGOutflow and SJOutflow time series slots in the JemezData data object. If so, the function result is set to the maximum of either the minimum Rio Grande outflow or the difference between the value in the Outflow time series slot in the Jemez storage reservoir object and the value in the SJOutflow time series slot in the JemezData data Object. If not, the function result is set to the value in the RGOutflow time series slot in the JemezData data object for the current timestep.

The value for the supply to transfer water from the MRGCD account at Jemez Reservoir to the destination of MRGCD\_MiddleValley as identified with the user-defined AllSuppliesByReleaseTypeDestination function in the Account Lists utility group is set using the user-defined ReconcileSJOutflow function in the General Account Functions utility group. This function computes the result as the value for the Outflow time series slot in the Jemez storage reservoir object for the current timestep minus the result from the ReconcileRGOutflow function.

#### **A.21.8. CheckJemezRGOutflow**

*Explanation:* Aborts the simulation if the supply for the release of Rio Grande water from the Rio Grande account at Jemez Reservoir was not set. This rule helps with debugging because a simulation will stop if the Rio Grande supply was not set and the problem and any missing input can be identified by backtracking in the calculations from the point in the simulation that the run aborted.

*Rule Execution:* There are no execution constraints for this rule.

*Rule Logic:* If the value for the Supply at Jemez Reservoir with a release type of Rio Grande and destination of RioGrande\_RioGrande is a NaN, the STOP\_RUN command is used to stop the simulation, and the PRINT statement is used to print a comment to the Diagnostics Output window that the run stopped because the account slot value was not set.



### **A.21.9. ReconciledRGandSJReleases**

*Explanation:* This rule records results for the reconciled Rio Grande and San Juan outflow from Heron, El Vado, Abiquiu, and Cochiti Dams to the data object for each reservoir.

*Rule Execution:* If the value for the ReconciledRGOutflow time series slot in the AbiquiuData data object for the current timestep is a NaN, the rule fires.

*Rule Logic:* Two FOREACH loops are used to set the values for the ReconciledRGOutflow and ReconciledSJOutflow time series slots in the HeronData, ElVadoData, AbiquiuData, and CochitiData data objects. The values are computed with the user-defined ReconcileRGOutflow and ReconcileSJOutflow functions in the General Account Functions utility group. Different equations are used in the ReconcileRGOutflow function depending on the reservoir. For the user-defined ReconcileSJOutflow function, the result is the value for the Outflow time series slot in the corresponding reservoir object for the current timestep minus the result from the ReconcileRGOutflow function for the corresponding reservoir object.

## **A.22. Preparatory ElephantButte Caballo**

The rules in this policy group are used to set the transfer of Compact credit water to the Rio Grande account that may result from relinquished credits and determine the percentage of the full demand out of Elephant Butte Dam that can be met and then used for setting releases from Elephant Butte and Caballo Dams.

### **A.22.1. NM Relinquishments**

*Explanation:* This rule sets the value for the transfer of water out of the NMCredit storage account to the Rio Grande account at Elephant Butte Reservoir based on any relinquished credits that are input or calculated.

*Rule Execution:* This rule fires if the switch is set to model relinquished credits as checked with reference to the user-defined ModelRelinquishedCredits function and the current timestep is equal to the input date for a relinquishment as checked with reference to the user-defined RelinquishmentDate function and the value for the RelinquishedNMCredits time series slot in the ElVadoData data object for the current timestep is greater than zero.

*Rule Logic:* A GET STRING @INDEX statement is used with the user-defined AllSuppliesIntraByReleaseTypeDestination function in the Account Lists utility group to identify the name of the supply for the assignment, and the value is set to the daily flow

for the input volume in the RelinquishedNMCredits slot in the ElVadoData data object for the current timestep plus the value for Transfers Out for the NMCredit account at Elephant Butte Reservoir for the current timestep.

### **A.22.2. Determine Percentage**

*Explanation:* This rule computes the percentage of the full demand below Elephant Butte that can be met with the available supply in storage and the forecasted inflow. Releases from Elephant Butte and Caballo dam are then set with reference to this computed percentage.

*Rule Execution:* The rule fires if it is the start timestep as determined with the user-defined IsStartOfRun function in the General Boolean Functions utility group or if it is the first day of a month as determined with the user-defined IsFirstDayOfMonth function in the Date Functions utility group and the current timestep is in May or earlier.

*Rule Logic:* The value for the DownstreamDemandPercentage time series slot in the ElephantButteData data object for the timestep at the end of the current month is set to the Storage in the Rio Grande account at Elephant Butte Reservoir at the previous timestep as identified with the user-defined PreviousAccountStorage function in the General Account Functions utility group plus the value in the Storage time series slot in the Caballo storage reservoir object at the previous timestep as identified with the user-defined PreviousStorage function in the Get Data Functions utility group plus the quantity of the value in the OtowiForecast time series slot in the Indian data object for December 31 of the current year unless it is a NaN, then the value in the AverageOtowiForecast scalar slot in the Indian data object, multiplied by the quantity of 1.0 minus the value in the ForecastLossPercentage table slot in the ElephantButteData data object all divided by the sum of the values in the AverageDemand periodic slot in the ElephantButteData data object for the current year as summed using the user-defined SumPeriodicSlot function in the Account Summing Functions utility group. The result is restricted to be no greater than 1.0 using the predefined Min function. Refer to Figure A.53 for a flowchart showing the logic used for this rule.

### **A.22.3. ElephantButte Downstream Demand**

*Explanation:* This rule computes the demand downstream from Elephant Butte Dam based on the computed downstream demand percentage and input average demand.

*Rule Execution:* The rule fires if it is the start timestep as determined with the user-defined IsStartOfRun function in the General Boolean Functions utility group or if it is the first day of a month as determined with the user-defined IsFirstDayOfMonth function in the Date Functions utility group and the current timestep is in May or earlier.

*Rule Logic:* A FOREACH loop is used to set the values in the DownstreamDemand time series slot in the ElephantButteData data object for the current timestep to December 31 of the current year or the finish timestep if it is earlier. The values are set to the value in the AverageDemand periodic slot in the ElephantButteData data object for each corresponding date multiplied by the value in the DownstreamDemandPercentage time series slot in the ElephantButteData data object for the timestep at the end of the current month. Refer to Figure A.53 for a flowchart that depicts the logic used in this rule.

#### **A.22.4. Caballo Downstream Demand**

*Explanation:* This rule computes the demand downstream from Caballo Dam using the same equation used for Elephant Butte Dam except the different average demand values input to the CaballoData data object are referenced and the results are recorded to the CaballoData data object. A flowchart for this rule is portrayed in Figure A.53.

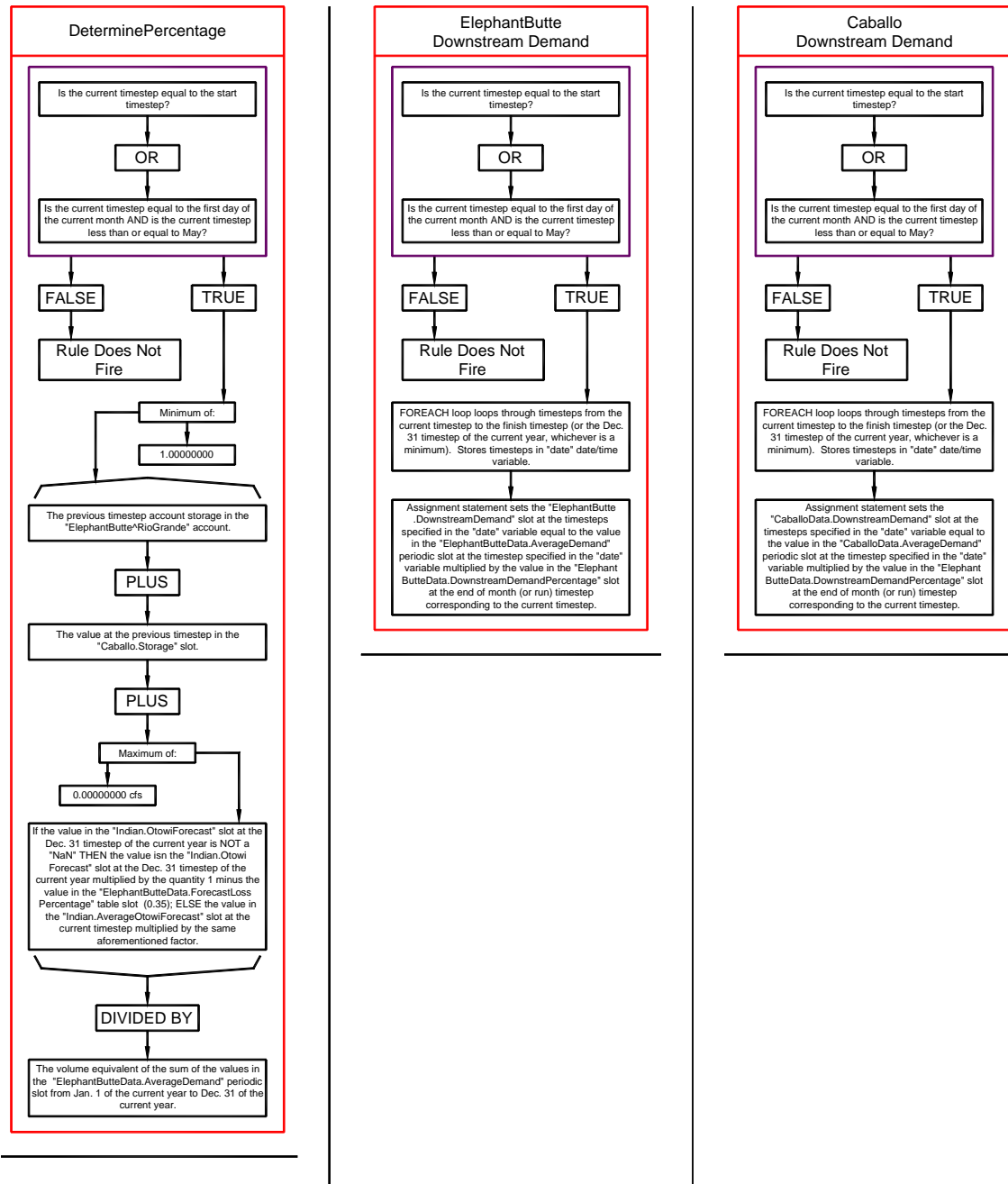


Figure A.53. Flowcharts for the Determine Percentage, Elephant Butte Downstream Demand, and Caballo Downstream Demand Rules

## **A.23. Elephant Butte**

The rules in this policy group are used to determine the outflow from Elephant Butte Dam with consideration for the downstream demand, the release to return to the prudent storage level, and downstream channel capacity.

### **A.23.1. ElephantButteOutflow**

*Explanation:* Unless a total outflow for Elephant Butte Dam has been input for the current timestep, the value is set to the computed downstream demand. The resulting total release is checked to see if it is physically legitimate based on the outlet works, and if not, the release is reset to reflect the restriction for the given outlet works.

*Rule Execution:* If the value for the Outflow time series slot in the ElephantButte level power reservoir object for the current timestep is a NaN, the rule fires.

*Rule Logic:* This rule contains an IF THEN ELSE statement to see if the value in the TotalOutflow time series slot in the ElephantButteData data object for the current timestep as referenced using the user-defined TotalOutflow function in the Get Data Functions utility group is a NaN. If not, the value for the Outflow time series slot in the ElephantButte level power reservoir object for the current timestep is set to that input TotalOutflow value. Otherwise, the value for the Outflow slot is set to the value in the DownstreamDemand time series slot in the ElephantButteData data object for the current timestep as referenced using the user-defined DownstreamDemands function in the Get Data Functions utility group. This function references the user-defined CheckReservoirOutflow [sic] function in the Caballo Functions utility group as a maximum constraint to assure the result does not exceed the maximum outflow as determined with the predefined GetMaxOutflowGivenInflow function. Within the rule, the result for the Outflow is checked against the physical restrictions of the outlet works using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1).

### **A.23.2. ElephantButteOutflowRestrictions**

*Explanation:* This rule resets the outflow from Elephant Butte Dam if the predetermined value exceeds the maximum by 10 cfs or a larger release is necessary to return to the prudent pool elevation.

*Rule Execution:* This rule fires if the rule has not already fired for the current timestep as checked with reference to the predefined HasRuleFiredSuccessfully function.

*Rule Logic:* The rule contains an IF THEN statement to see if the value for the Outflow time series slot in the ElephantButte level power reservoir object minus the result from

the user-defined ElephantButteComputedMaxOutflow function in the Elephant Butte Functions utility group is greater than 10 cfs or the result from the user-defined EBPrudentRelease function in the Elephant Butte Functions utility group is greater than the value in the DownstreamDemand time series slot in the ElephantButteData data object for the current timestep as referenced using the user-defined DownstreamDemands function in the Get Data Functions utility group. If those criteria are satisfied, the value for the Outflow time series slot in the ElephantButte level power reservoir object is reset to the result from the ElephantButteComputedMaxOutflow function. This function sets the outflow to the minimum of the channel capacity as input to the ChannelCapacities table slot in the ElephantButteData data object and referenced using the user-defined ChannelCapacity function in the General Channel Capacity Functions utility group which is referenced in the user-defined ElephantButteReleaseForDSChannelCapacity function in the Elephant Butte Functions utility group or the result from the EBPrudentRelease function or the result from the EBReleaseForCaballo function.

The EBPrudentRelease function includes an IF THEN ELSE statement to see if it is NOT the irrigation season as determined using the user-defined NonIrrigationSeason function in the Date Functions utility group and if the expected elevation at Elephant Butte Reservoir for the current inflow as determined with predefined StorageToElevation function within the user-defined StorageToElevationBasedOnCurrentInflow function in the General Compute Functions utility group is greater than the storage space between the top of the conservation pool as input to the PoolLevels table slot in the ElephantButteData data object and the WINTER value input to the PrudentAvailableStorage table slot in the ElephantButteData data object. If those criteria are satisfied, the function result is set to the outflow required to reduce the pool elevation below the WINTER prudent storage level as computed with the user-defined OutflowToGetBelowPrudentPool function in the General Compute Functions utility group and restricted to be no less than the value in the DownstreamDemand time series slot in the ElephantButteData data object for the current timestep as referenced using the user-defined DownstreamDemands function. Otherwise, if it is the irrigation season (March 1 through October 31) as determined with the user-defined IrrigationSeason function in the Date Functions utility group and if the expected elevation at Elephant Butte Reservoir for the current inflow is greater than the storage space between the top of conservation pool and the SUMMER value in the PrudentAvailableStorage table slot in the ElephantButteData data object. If not, the function result is set to the value in the Outflow time series slot in the ElephantButte level power reservoir object for the current timestep. If so, the function result is set to the outflow to reduce the pool elevation below the SUMMER prudent storage level as computed with the user-defined OutflowToGetBelowPrudentPool function and restricted to be no less than the value in the DownstreamDemand time series slot in the ElephantButteData data object for the current timestep.

Several criteria are checked in the EBReleaseForCaballo function, and depending on whether the criteria are satisfied, the result is set to either be 0.0 cfs or the channel capacity as input to the ChannelCapacities table slot in the ElephantButteData data object.

## **A.24. Elephant Butte Accounts**

The rule in this policy group sets the supply for the release of Rio Grande water and Albuquerque water from Elephant Butte Dam.

### **A.24.1. SetElephantButteAccounts**

*Explanation:* The value for the supply to transfer water from the Rio Grande account downstream is set to the outflow from Elephant Butte Dam, and the value for the supply to transfer water from the Albuquerque account downstream is set to 0.0 cfs.

*Rule Execution:* If the RioGrandeElephantButteToRioGrandeBlwElephantButte supply for the current timestep is a NaN, the rule fires.

*Rule Logic:* This rule includes two assignment statements. The value for the Supply for the RioGrandeElephantButteToRioGrandeBlwElephantButte transfer is set to the value in the Outflow time series slot in the ElephantButte level power reservoir object for the current timestep. The value for the Supply for the AlbuquerqueElephantButteToAlbuquerqueBlwElephantButte transfer for the current timestep is set to 0.0 cfs.

## **A.25. Caballo Flood Control Rules**

The rules in this policy group are used to set the outflow from Caballo Dam and check the resulting outflow against flood control criteria, the downstream channel capacity, and the physical constraints of the outlet works.

### **A.25.1. CaballoOutflow**

*Explanation:* Unless a total outflow for Caballo Dam has been input for the current timestep, the value is set to the computed downstream demand or the outflow to drop below the flood pool. The resulting total release is checked to see if it is physically legitimate based on the outlet works, and if not, the release is reset to reflect the restriction for the given outlet works.

*Rule Execution:* If the value for the Outflow time series slot in the Caballo storage reservoir object for the current timestep is a NaN, the rule fires.

*Rule Logic:* This rule contains an IF THEN ELSE statement to see if the value in the TotalOutflow time series slot in the CaballoData data object for the current timestep as referenced using the user-defined TotalOutflow function in the Get Data Functions utility

group is a NaN. If not, the value for the Outflow time series slot in the Caballo storage reservoir object for the current timestep is set to that input TotalOutflow value. Otherwise, the value for the Outflow slot is set to the maximum of two values: 1) the value in the DownstreamDemand time series slot in the CaballoData data object for the current timestep as referenced using the user-defined DownstreamDemands function in the Get Data Functions utility group. This function references the user-defined CheckReservoirOutflow [sic] function in the Caballo Functions utility group as a maximum constraint to assure the result does not exceed the maximum outflow as determined with the predefined GetMaxOutflowGivenInflow function. 2) The outflow to drop below the flood pool as determined with the user-defined OutflowToGetBelowFloodPool function in the General Compute Functions utility group which references the predefined SolveOutflow function is the other value.

Within the rule, the result is checked against the physical restrictions of the outlet works using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1).

## **A.25.2. CaballoOutflowRestrictions**

*Explanation:* This rule resets the outflow from Caballo Dam to comply with the downstream channel capacity at El Paso if necessary. The resulting total release is checked to see if it is physically legitimate based on the outlet works, and if not, the release is reset to reflect the restriction for the given outlet works.

*Rule Execution:* The rule fires if the rule has not already fired for the current timestep as checked with reference to the predefined HasRuleFiredSuccessfully function and if the value for the Outflow time series slot in the Caballo storage reservoir object for the current timestep minus the result from the user-defined CaballoChannelCapacity function in the CaballoFunctions utility group is greater than the value of the Tolerance at ElPaso input to the ChannelCapacities table slot in the CaballoData data object.

*Rule Logic:* If the execution constraints are satisfied, the value for the Outflow time series slot in the Caballo storage reservoir object for the current timestep is set to the result from the user-defined CaballoChannelCapacity function in the CaballoFunctions utility group. This function uses the predefined HypTargetSimWithStatus function to determine the outflow from Caballo Dam (or the value for the Gage Inflow time series slot in the BlwCaballo stream gage object) to meet a target flow at El Paso (or the Gage Inflow time series slot in the ElPaso stream gage object) at the timestep equal to the current timestep plus the number of timesteps input to the ApproxNoOfDaysDS table slot in the CaballoData data object. The target is the channel capacity at ElPaso as input to the ChannelCapacities table slot in the CaballoData data object. The function completes the hypothetical simulation for all the objects in the BlwCaballoToElPaso subbasin and considers a check against a minimum outflow from Caballo Dam of 0.0 cfs and a maximum outflow equal to the result from the user-defined ComputeMaxOutflow function in the General Compute Functions utility group which references the predefined



GetMaxOutflowGivenInflow function. The tolerance is equal to the input value in the ChannelCapacities table slot in the CaballoData data object.

Within the rule, the result is checked against the physical restrictions of the outlet works using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.12.2.1).

### **A.25.3. CaballoFloodControlReleaseBasedOnFORD**

*Explanation:* This rule is used to reset the outflow from Caballo Dam if a flood release is required to reduce the elevation below the flood pool elevation.

*Rule Execution:* The rule fires if the value for the FloodRelease time series slot in the CaballoData data object for the current timestep is a NaN and if the value for the Outflow time series slot in the Caballo storage reservoir object is NOT a NaN and if the result from the user-defined IsFloodReleaseRequired function is true. This function checks to see if the anticipated elevation as determined with the predefined StorageToElevation function and with the current inflow and predetermined outflow is greater than the BottomOfFlood elevation in the PoolLevels table slot in the CaballoData data object.

*Rule Logic:* If the execution constraints are satisfied, this rule includes two assignment statements. The value in the Outflow time series slot in the Caballo storage reservoir object for the current timestep and the value in the FloodRelease time series slot in the CaballoData data object are both set to the result from the user-defined CaballoFloodReleaseBasedOnFORD function in the Caballo Functions utility group. This function includes an IF THEN ELSE statement to see if the anticipated elevation based on the current inflow and predetermined outflow as computed with the user-defined StorageToElevationBasedOnCurrentInflow function in the General Compute Functions utility group which references the predefined StorageToElevation function is greater than the Max elevation in the PoolLevels table slot in the CaballoData data object. If so, the outflow is set to the result from the user-defined ComputeMaxOutflow function in the General Compute Functions utility group which references the predefined GetMaxOutflowGivenInflow function. Otherwise, the function result is set to the maximum of the result from the user-defined ComputeCaballoFloodReleaseBasedOnFORD function in the Caballo Functions utility group or the value in the DownstreamDemand time series slot in the CaballoData data object for the current timestep as referenced with the user-defined DownstreamDemands function in the Get Data Functions utility group. This function references the user-defined CheckReservoirOutflow [sic] function in the Caballo Functions utility group as a maximum constraint to assure the result does not exceed the maximum outflow as determined with the predefined GetMaxOutflowGivenInflow function.

The ComputeCaballoFloodReleaseBasedOnFORD function checks to see if the pool elevation in the Caballo storage reservoir object at the previous timestep is greater than the BottomOfFlood elevation in the PoolLevels table slot in the CaballoData data object

and if the value in the Outflow time series slot in the Caballo storage reservoir object at the previous timestep is equal to 5000 cfs, the downstream channel capacity, and if the value in the Inflow time series slot in the Caballo storage reservoir object for the current timestep is less than or equal to the previous inflow plus 1000 cfs. If those criteria are satisfied, the function result is set to the previous outflow. If not, a second IF THEN ELSE statement is used to see if the result from the IsFloodReleaseRequired function is true given the current inflow and an outflow looked up in the FloodReleaseTable table slot in the CaballoData data object. If so, the function result is set to that value looked up in the FloodReleaseTable. If not, the function result is set to the minimum of the current inflow, the value from the FloodReleaseTable, or the result from the user-defined OutflowToGetBelowFloodPool function in the General Compute Functions utility group.

For both assignments within the rule, the result is checked against the physical restrictions of the outlet works using the user-defined CheckThisResPhysicalConstraints function (Refer to Section A.11.2.1).

## **A.26. HeronRioGrandeStorageAdjustment**

The rule in this policy group is used for the end-of-year accounting adjustment at Heron Reservoir to offset for impacts of Rio Grande water in storage on losses to the San Juan-Chama Project water.

### **A.26.1. HeronRioGrandeAdjustment**

*Explanation:* On January 1, this rule sets the supply for the previous timestep to transfer 350 acre-ft from the Rio Grande account to the Federal San Juan account at Lake Heron to offset for recreation and loss impacts due to storage of Rio Grande water on the supply of San Juan-Chama Project water.

*Rule Execution:* On January 1 of each year of simulation but not the start timestep, the rule fires.

*Rule Logic:* The value for the Supply for the RioGrandHeronToFederalSanJuanHeron transfer for the previous timestep is set to the flow corresponding to transferring a volume of 350 acre-ft over one day as computed with the predefined VolumeToFlow function.