Modeling the Broad River Basin Operations with OASIS

Addendum to the
User Manual for OASIS with OCL™

December 2011
Revised April 2012

Prepared for the

North Carolina Division of Water Resources

HYDROLOGICS
Advancing the management of water resources

Table of Contents

Section 1. Introduction

This report describes how OASIS is used to model the operations of the Broad River Basin in North Carolina. This application of OASIS, known as the Broad River Basin Hydrologic Model (the model), extends geographically from the headwaters of the Broad River to the river's confluence with Buffalo Creek. The confluence is the most downstream point where the Broad River is impacted by operations in North Carolina. This report is not intended to replace the User Manual for OASIS, which is available from the Help menu of the model. Rather, it is intended to document the data used in this application as well as the current operations of the basin. Information about the OASIS platform is included only to the extent necessary to provide context for the application-specific data.

The model is available for registered users on the Division of Water Resources (DWR) server. The model can be used in two modes: (1) a simulation mode to evaluate system performance for a given set of demands, operating policies, and facilities over the historic inflow record; and (2) a position analysis mode for real-time management. In the latter mode, the model uses multiple ensemble forecasts to provide a probabilistic assessment of conditions up to one year in the future. Although it can be used for other purposes as well, this feature is particularly useful for drought management. The simulation mode contains three default runs, one for conditions today, one for projected 2030 conditions, and one for conditions at the end of the 50-year planning horizon in 2060.

The model uses an inflow data set that extends from January 1, 1929 through September 30, 2009. This data set was developed using a comprehensive approach that (1) relies on eight streamflow gages in the basin as well as four outside of the basin; (2) accounts explicitly for upstream alterations, or impairments, from reservoir regulation and net water consumption; and (3) uses statistical techniques to complete missing records for these gages.

Real-time drought management depends upon having current forecasts of inflow. As noted below, the generation of such forecasts is dependent upon having inflows through the present day. Updating the inflows requires the collection of impairment data, which can be time intensive. It is envisioned that these data will be collected every five years. In the interim (e.g., through 2014), the inflow data starting October 2009 will be based on a provisional inflow technique so that real-time updates can be made quickly and easily without the need to collect all the impairment data.

The remainder of this document summarizes the components of the model and the major operations in the basin. Appendix A lists the static input data and run code used in the basecase simulation run that is based on today's facilities, operations, and demands. Appendix B describes the approach used to establish the finalized inflow data set. Verification of inflows and operating rules was described in meetings with the Technical Review Committee and formalized in Powerpoint® presentations that are available on the DWR website. Appendix C describes the approach for generating provisional inflows.

Appendix D describes the weighting assigned to the various nodes and arcs so that the model reflects the general priorities for water allocation in the basin.

It is important to note how the OASIS model should and should not be used. OASIS is a generalized type of mass balance model used mainly in evaluating planning and management alternatives. It is not intended for use in hydraulic routing nor flood management, although it can be linked to other models for those purposes.

In addition, since modeling results are sensitive to inflows, the user must be cautioned about accuracy of the inflows. HydroLogics spent considerable effort in developing the inflow data. The methodology ensures that the monthly naturalized flows at the gage locations match, which assumes that any measurement error is embedded in the impairments and not the streamflow data. DWR agreed to this method, which, although imperfect, is the most reasonable given the available data. Further, it is important to note that we are not trying to replicate history in computing the OASIS inflows; rather, we are trying to build a data set of daily flows whose variation is *representative* of history while preserving monthly gaged flows as "ground truth".

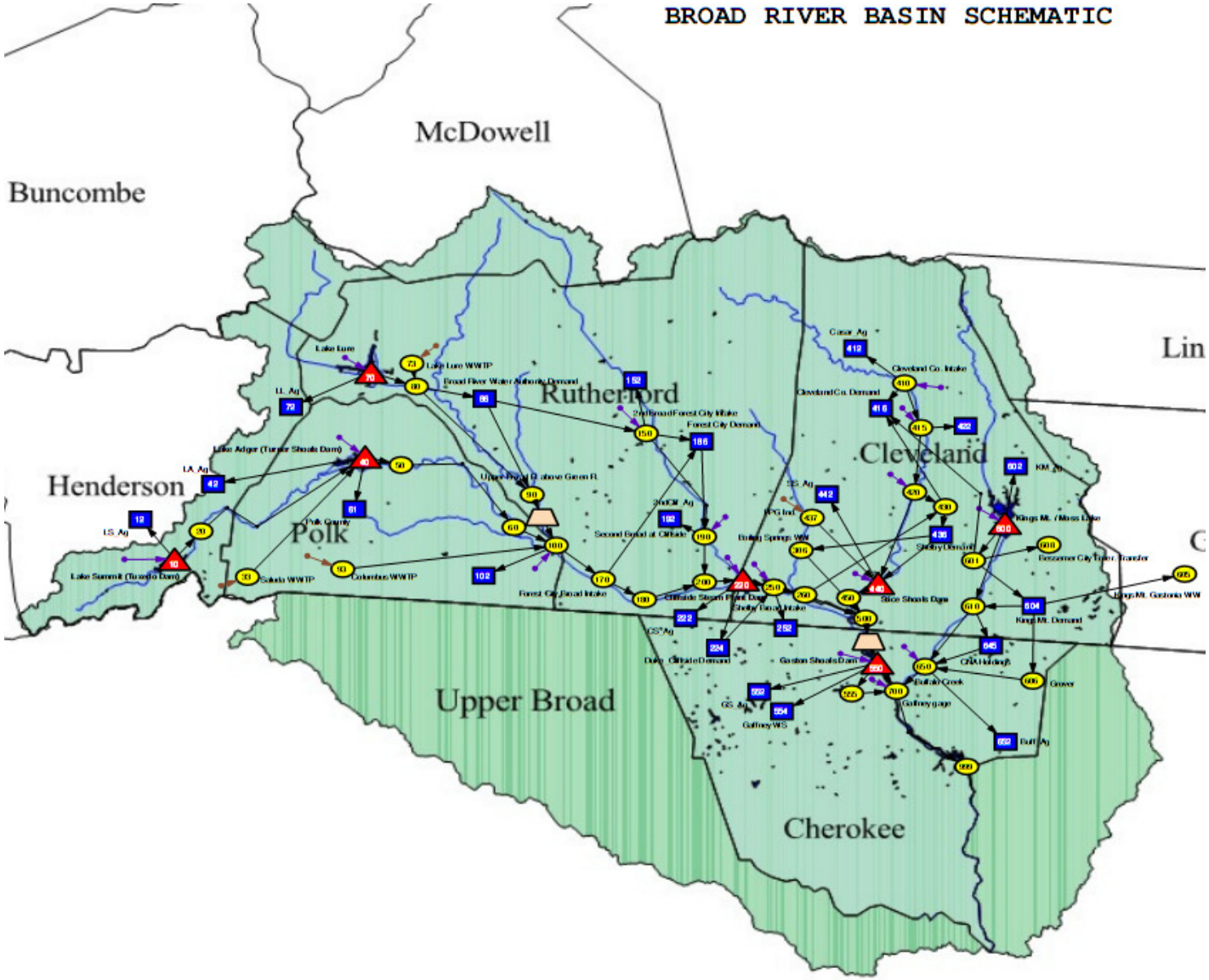
Due partly to the inaccuracy of some of the impairment data and to time of travel, negative inflows may occur. These can lead to potential model infeasibility. The model code filters out negative inflows, particularly large ones, but preserves the total inflow volume over a short period by debiting those negative inflows from subsequent positive inflows. For example, if a rainstorm hits the upstream part of the reach but not the downstream part, the gaged flow data may indicate a large negative inflow (gain) between the upstream and downstream ends of a stream reach. When the flow attenuates upstream and peaks downstream, the inflow becomes positive, and the negative gain from the day(s) before is (are) debited from the positive inflows the day(s) after to ensure that the average inflow over that period is preserved. The occurrence of negative inflows is reduced in the main-stem of the Broad by incorporating time-of-travel equations recommended by DWR. These equations are provided in Appendix B.

Section 2. Model Components

2.1 Schematic

The model uses a map-based schematic that includes nodes for withdrawals (agricultural, municipal, and industrial), discharges (municipal and industrial), reservoirs, and inflows, and arcs that represent means of water conveyance between nodes. The model schematic is shown on the following page and is sized to show the full system. (To make the schematic more legible, the user can adjust the schematic size from the model's graphical user interface (GUI)). The schematic and associated physical data were developed with input from basin stakeholders at numerous model review meetings.

In total, the model has approximately 70 nodes and 80 connecting arcs. There are 9 reservoir nodes (seven actual reservoirs and two artificial storage nodes used for time-of-travel flow routing), 14 irrigation nodes, 9 municipal and industrial demand nodes, 4 independent discharge nodes, 16 natural inflow nodes (including the reservoir nodes), and other miscellaneous nodes to account for minimum flow requirements, interconnections, and instream flow assessment for ecological needs.


BROAD RIVER BASIN SCHEMATIC
McDowell
Buncombe
Rutherford
Cleveland
Henderson
Polk
Upper Broad
Cherokee

The user can click on any node or connecting arc on the schematic to access specific information, like reservoir elevation-storage-area data or minimum streamflow requirements. These data are also contained in tables contained on other tabs of the model.

2.2 Model Input

Input data for the model is stored in three forms: static and pattern data, timeseries data, and user-defined data using operations control language (OCL). The timeseries data are stored outside the model run. The other data are embedded in the run and copy over automatically when creating a new run.

Static and pattern data are contained in the GUI and represent data that do not change during the model simulation (such as physical data like reservoir elevation-storage-area relationships) or repeating data that occurs every year in the simulation (like monthly demand patterns or seasonal minimum release patterns). Timeseries data change with each day in the simulation record and typically consist of inflows and reservoir net evaporation. OCL allows the user to define more elaborate operating rules than are permitted from the GUI.

Static and Pattern Data

Tables containing the model's static and pattern data can be found in Appendix A. Reservoir information includes elevation-storage-area relationships, minimum and maximum allowable storage, and any rule curves which dictate the preferred operating elevation throughout the year.

Minimum flows and reservoir releases are defined by minimum flow patterns on arcs.

Water treatment plant and transmission constraints are defined by maximum capacities on arcs.

Municipal and industrial demand nodes use an annual average demand subject to a monthly pattern, and an associated wastewater discharge based on a fraction of the monthly demand. Wastewater discharges not associated with demand nodes are modeled using an annual average return subject to a monthly pattern.

The model allows the user to systematically adjust all municipal and industrial demands in the basin by invoking the demand multiplier option on the Setup tab. This is useful when doing sensitivity analyses on the impact of demand growth in the basin. Note that irrigation demands and independent wastewater returns are not adjusted using this multiplier but can be updated using the procedure described in the next section. The few independent wastewater returns in the model can be adjusted manually in the pattern tables.

Timeseries Data

The timeseries data are stored in a basedata timeseries file (*basedata.dss*), which contains all the inflow and net evaporation (evaporation less precipitation) data. The sources for these data are provided in Appendix B along with a more detailed description of how the inflows were developed. As noted, updating the timeseries data can be done in two ways: (1) using the comprehensive approach described in Appendix B; or (2) using the provisional approach for facilitating real-time drought management described in Appendix C. The provisional approach relies on data from select gaging and precipitation stations throughout the basin. The provisional updates can be done directly from the interface by selecting the Update Record tab, inputting the data, and clicking on the Update Record button. The update record algorithm will calculate the inflows to all the OASIS inflow nodes and net evaporation for all reservoir nodes and write them to the *basedata.dss* file automatically.

Agricultural water use is modeled as a timeseries over the historic hydrologic record. It is broken out by county and depends on livestock count, crop usage, livestock and crop water consumption, and rainfall. Evapotranspiration equations for each crop are used in conjunction with the timeseries precipitation record so that crops are only irrigated when necessary. The water use can be easily adjusted from the model interface by opening the Edit Irrigation Data dialog box on the Setup tab. The model automatically converts the input data on crop acreage and livestock count into water use values. The agricultural demand nodes are a summation of the agricultural water usage in a particular reach of interest.

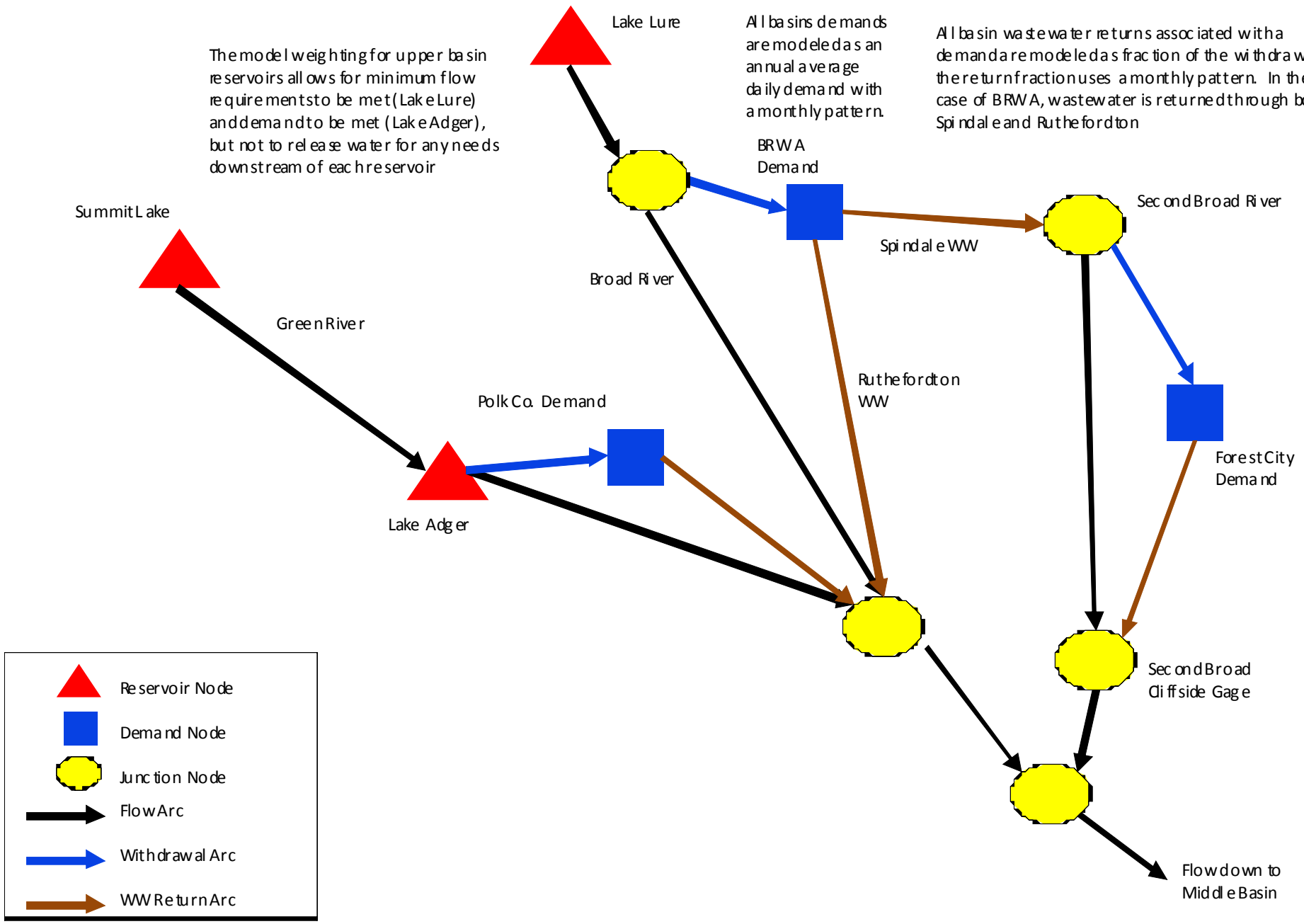
Operating Rules

As described in more detail in Appendix D, most of the water allocation priorities are set by the user in the GUI by applying weights to nodes and arcs. The most common operating rules are for storing water in reservoirs versus releasing the water to meet local demands or minimum releases, and these are reflected by the weighting scheme. Simply stated, if a minimum flow in a river is more important than meeting the local water supply demand, a higher weight on the minimum flow means water supply deliveries will be scaled back in order to meet the minimum flow.

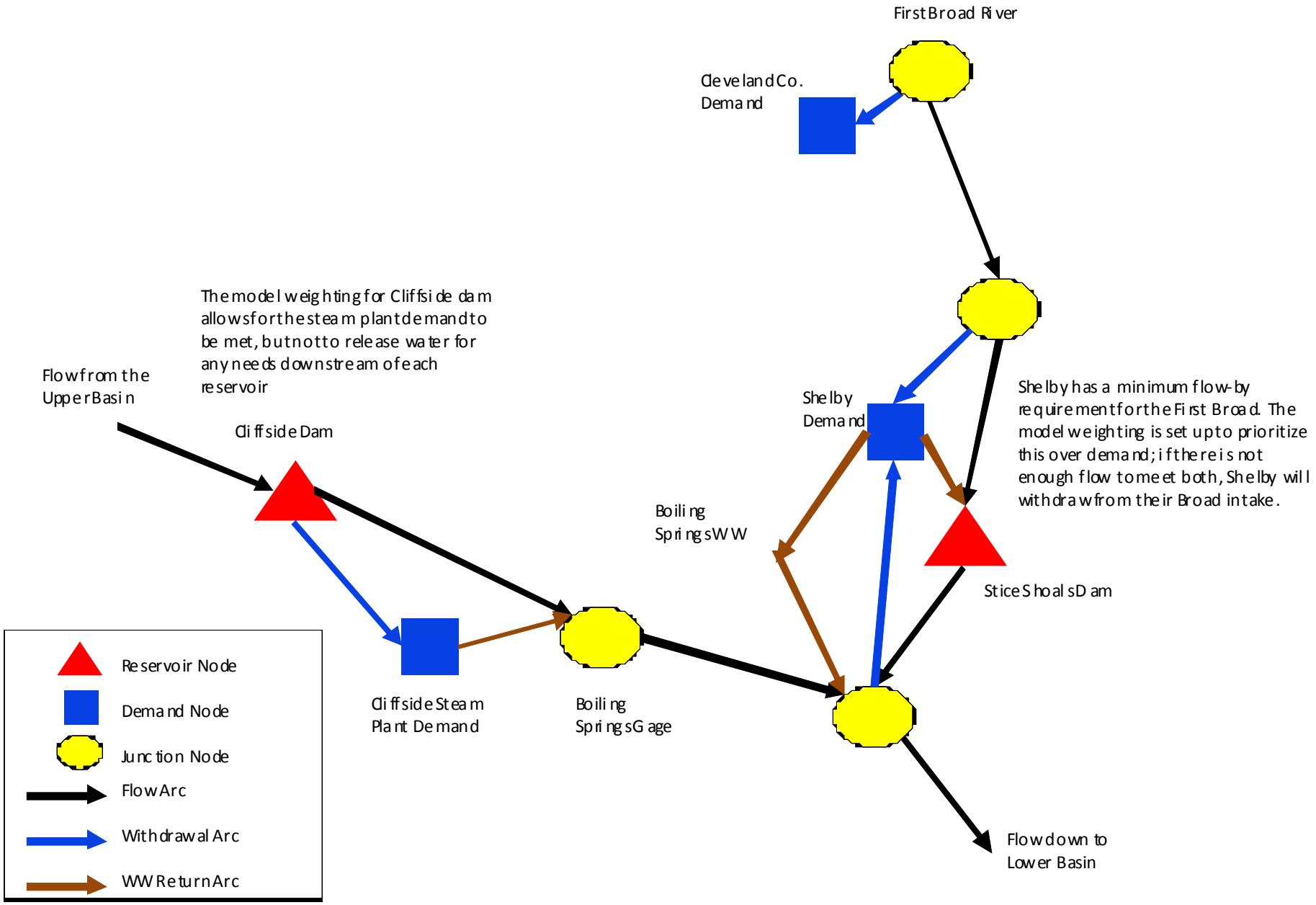
The Operations Control Language (OCL) allows the user to model more complex operating rules, such as tying demand reductions to reservoir levels or river flows that are common in drought management. These files are accessible from the model interface. The OCL files associated with the basecase simulation run that uses year 2010 demands are included in Appendix A. The key OCL files include *main.ocl*, which initializes the run and refers to all the other OCL files; i*nflows.ocl*, which filters the inflows for any negative gains in the provisional record; *return_flows.ocl* which sets the wastewater returns; *routing.ocl*, which routes water to account for time of travel; and *drought_plans.ocl* which codes the Water Shortage Response Plans submitted by utilities to DWR.

A flowchart is provided below summarizing the overall operations of the basin as captured in the model. Appendix A contains the OCL files, and detail on weighting is provided in Appendix D.

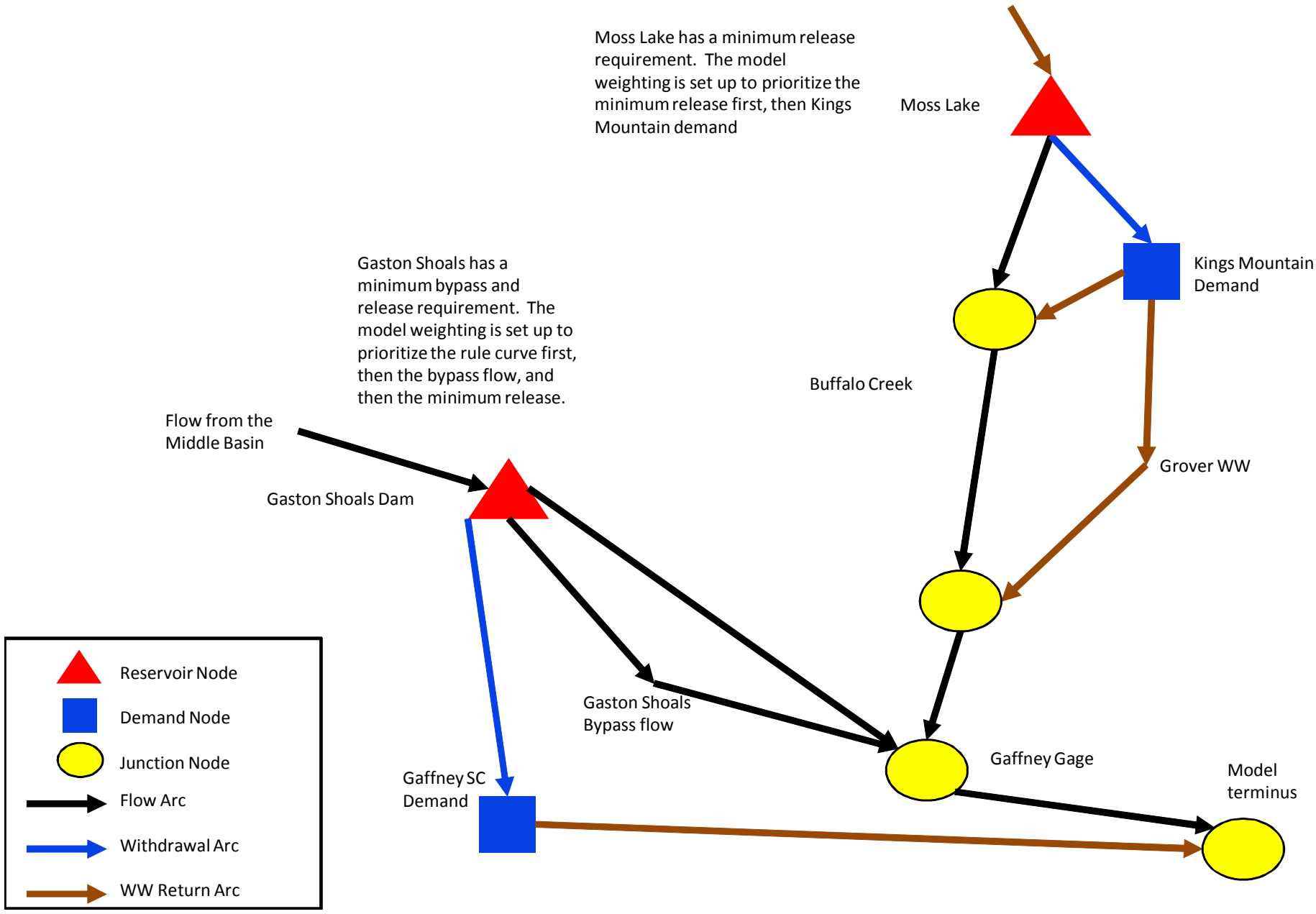
Flow Chart of Major Nodes in the Upper Broad Basin



Flow Chart of Major Nodes in the Middle Broad Basin



Flow Chart of Major Nodes in the Lower Broad Basin



2.3 Run Configurations

The model can be used in two modes: (1) a simulation mode to evaluate system performance for a given set of demands, operating policies, and facilities over the historic inflow record; and (2) a position analysis mode for real-time management. General information on creating, modifying, and executing runs is provided in the User Manual for OASIS, which is available from the Help menu of the model.

Simulation:

In simulation mode, on the Setup tab, the user can select from three radio buttons: No Forecasts, Conditional Forecasts, and Non-conditional Forecasts. The latter two enable the user to evaluate forecast-based operating policies (although none are used in the basecase scenario), with inflow forecasts generated for each week in the historical inflow record. Conditional forecasts account for antecedent flow conditions while non-conditional forecasts are made independent of how wet or dry the basin is. The forecasts for the simulation mode are generated outside the GUI and stored in the basedata folder. The current forecast file is developed from the timeseries *basedata.dss* file that extends through September 2009. The forecast file should only be updated in conjunction with the comprehensive inflow updates (anticipated every five years with the first update in 2014).

Position Analysis:

In position analysis mode, the user can select from Conditional or Non-Conditional Forecasts on the Setup tab. By executing a run, the model will produce a forecast (typically of river flows or reservoir elevations) for up to the next 365 days. A forecast can be made on any date in the historic inflow record or no more than one day after the end of the inflow record. Typically it will be used starting the day after the last update of the inflow and net evaporation record. For example, if these records end September 30, 2009, the user can run a forecast for October 1, 2009. If a month has passed, and the user wants to run a forecast for November 1, 2009, the user would update the inflows and net evaporation for October using the Update Record tab and then start the position analysis run on November 1. For a reservoir, or locations affected by the operation of a reservoir upstream, the forecast is dependent on the starting elevation of the reservoir. On the Setup tab, the user simply inputs the starting elevation (or storage), the starting date of the forecast run, and clicks Run.

2.4 Model Output

The model allows the user to customize output files (tables or plots) and save them for routine use. Alternatively, the user can click on any node or arc on the schematic or go to the Setup tab and select Quick View to access and save tabular or plotted output. The balance sheet can also serve as a useful tool for tracking water through the system.

Included in the model output tables is a file called *xQy_ClimaticYear_BSgage.1v*. This file allows the user to compute instream flow statistics, such as 7Q10, for a specific site, in this case the Boiling Springs gage. To generate statistics for a different site, the user would copy and rename the file, then change the name and associated arc listed in the file. When viewing the generated output, the default layout shows two columns, for 7- and 30-day low flows (these periods can be changed in the .1v file). Scrolling to the bottom of the output file shows Log Pearson percentiles for each column. If the user is interested in the 7Q10 (7-day low flow, 10^{th} percentile) flow, the user would look at the first column, and the row labeled LPrs.100.

In addition, the model is capable of automatically determining the safe yield for a specific demand node, in this case the demand from Moss Lake. To generate statistics for a different site, the user would copy and rename the file (currently called *SafeYield_MossLake.1v*), then change the name and associated demand node listed in the file. The safe yield can be determined for each year in the historic inflow record (annual safe yield analysis) or for the entire period of record. The user inputs the adjustment criteria by selecting the Run Safe Yield Analysis button on the Setup tab. The safe yield routine works by tracking demand shortages for the chosen demand node, and iteratively works towards the maximum demand that produces no shortages from the supply source (in this example, Moss Lake). Note that the drought plans cannot be activated when using the safe yield routine, as the demand reductions resulting from drought restrictions inherently produce a 'shortage' from the normal demand.

APPENDIX A –

Model Static Input Data and Run Code

[from basecase run called "SimBase" using current conditions]

Static data tables included are:

All Model Nodes
All Model Arcs
Reservoirs Nodes
Reservoir Rules
Reservoir SAE Data
Demand Patterns
Wastewater Return Patterns
Arc Minimum Flows
County Irrigation Data
Crop Irrigation Coefficients

OCL files included are:

main.ocl
udef_list.ocl
agric_calculation.ocl
agric_allocation.ocl
set_inflows.ocl
return_flows.ocl
routing.ocl
drought_plans.ocl

Model Nodes

Node Number	Type	Inflow	Name
010	Reservoir	OCL	Lake Summit (Tuxedo Dam)
012	Demand	None	LS_Ag
020	Junction	None	Lake Summit Outflow
033	Junction	Pattern	Saluda WWTP
040	Reservoir	OCL	Lake Adger (Turner Shoals Dam)
042	Demand	None	LA_Ag
050	Junction	None	Green River below Turner Shoals
060	Junction	None	Lower Green River at Ken Miller Bridge
061	Demand	None	Polk County
070	Reservoir	OCL	Lake Lure
072	Demand	None	LL_Ag
073	Junction	Pattern	Lake Lure WWTP
080	Junction	None	Lake Lure Outflow
086	Demand	None	Broad River Water Authority Demand
090	Junction	None	Upper Broad R. above Green R.
093	Junction	Pattern	Columbus WWTP
095	Reservoir	None	Upper Broad Routing
100	Junction	OCL	Green R. Confluence
102	Demand	None	BRoad_Green_Ag
150	Junction	OCL	2nd Broad Forest City Intake
152	Demand	None	Logn_Ag
170	Junction	None	Forest City Broad Intake
180	Junction	None	Broad R. upstream Cliffside Steam Plant
186	Demand	None	Forest City Demand
190	Junction	OCL	Second Broad at Cliffside
192	Demand	None	2ndClif_Ag
200	Junction	None	2nd Broad Conf.
220	Reservoir	OCL	Cliffside Steam Plant Dam
222	Demand	None	CS_Ag
224	Demand	None	Duke_Cliffside Demand
250	Junction	OCL	Boiling Springs Gage
252	Demand	None	BS_Ag
260	Junction	None	Shelby Broad Intake
306	Junction	None	Boiling Springs WW
410	Junction	OCL	Cleveland Co. Intake
412	Demand	None	Casar_Ag

Node Number	Type	Inflow	Name
415	Junction	OCL	Lawndale Gage
416	Demand	None	Cleveland Co. Demand
420	Junction	OCL	First Broad Shelby
422	Demand	None	Lawn_Ag
430	Junction	None	Shelby Total WD
436	Demand	None	Shelby Demand
437	Junction	Pattern	PPG Ind.
440	Reservoir	OCL	Stice Shoals Dam
442	Demand	None	SS_Ag
450	Junction	None	Broad_First Broad Confluence
500	Junction	None	Node 500
525	Reservoir	None	Lower Broad Routing
550	Reservoir	OCL	Gaston Shoals Dam
552	Demand	None	GS_Ag
554	Demand	None	Gaffney WS
555	Junction	None	GS Bypass
600	Reservoir	OCL	Kings Mt. / Moss Lake
601	Junction	None	Kings Mt. Total WD
602	Demand	None	KM_Ag
604	Demand	None	Kings Mt. Demand
605	Junction	None	Kings Mt. Gastonia WW
606	Junction	None	Grover
608	Junction	None	Bessemer City Emer. Transfer
610	Junction	None	Kings. Mt. Outflow
645	Demand	None	CNA Holdings
650	Junction	OCL	Buffalo Creek
652	Demand	None	Buff_Ag
700	Junction	OCL	Gaffney gage
999	Junction	None	Terminal

Model Arcs

U/S Node	D/S Node	Name	Min Flow	Max Flow
010	012	LS_Ag WD	None	None
010	020	Lake Summit Release	None	None
020	040	Green R. to L. Adger	None	None
033	040	Saluda WW	None	None
040	042	LA_Ag WD	None	None
040	050	Lake Adger Release	None	None
040	061	Polk Co. WD	None	None
050	060	Green R. to Ken Miller Br	None	None
060	100	Green R. to Broad Conf.	None	None
070	072	LL_Ag WD	None	None
070	080	Lake Lure Release	Pattern	None
073	080	Lake Lure WW	None	None
080	086	Broad R. WA WD	None	None
080	090	80>90	None	None
086	090	Ruthefordton WWTP	None	None
086	150	Spindale WWTP	None	None
090	095	90>95	None	None
093	100	Columbus WW	None	None
095	100	Broad R. to Green R. Conf	None	None
100	102	100>102	None	None
100	170	100>170	None	None
150	152	150>152	None	None
150	186	Forest City WD	None	None
150	190	2nd Broad to Cliffside	None	None
170	180	170>180	None	None
170	186	Forest City Broad WD	None	None
180	200	180>200	None	None
186	190	Forest City WW	None	None
190	192	Nd5_Ag WD	None	None
190	200	2nd Broad to Broad Conf	None	None
200	220	U/S Cliffside Plant	None	None
220	222	CS_Ag WD	None	None
220	224	Duke_Cliffside WD	None	None
220	250	D/S Cliffside Plant	None	None
224	250	Cliff WW	None	None
250	252	250>252	None	None

U/S Node	D/S Node	Name	Min Flow	Max Flow
250	260	Boiling Springs Gage Flow	None	None
260	430	Shelby Broad WD	None	Pattern
260	500	260>600	None	None
306	500	Boiling Springs WW	None	None
410	412	Nd6_Ag WD	None	None
410	415	410>415	None	None
410	416	CCW First Broad WD	None	None
415	420	415>420	None	None
415	422	415>422	None	None
420	430	Shelby First Broad WD	None	None
420	440	Lower 1st Broad	Pattern	None
430	416	Shel>CCW	None	Pattern
430	436	Shelby WS WD	None	None
436	306	Shelby>Boil	None	None
436	440	Shelby WW	None	None
437	440	PPG WW	None	None
440	442	SS_Ag WD	None	None
440	450	Stice Shoals Outflow	None	None
450	500	1st Broad to Broad Conf.	None	None
500	525	500>525	None	None
525	550	525>550	None	None
550	552	GS_Ag WD	None	None
550	554	Gaffney Water Supply	None	None
550	555	GS Bypass1	Pattern	None
550	700	Gaston Shoals Release	OCL	None
555	700	GS Bypass2	None	None
600	601	Kings. Mt. Total WD	None	None
600	602	KM_Ag WD	None	None
600	610	Kings Mt. Release	Pattern	None
601	416	Clev. Co. Emer. Trans.	None	Pattern
601	604	Kings Mt. WS WD	None	None
601	608	Bessemer Emer. Trans.	None	Pattern
604	605	Kings Mt. Gastonia WW	None	Pattern
604	606	KM>Grover	None	None
604	610	Kings Mt. WW	None	None
606	650	Grover WW	None	None
610	645	CNA WD	None	None

U/S Node	D/S Node	Name	Min Flow	Max Flow
610	650	Buffalo Ck.	None	None
645	650	CNA WW	None	None
650	652	Nd9_Ag WD	None	None
650	700	Buffalo Ck. to Broad Conf	None	None
700	999	System Outflow	None	None

Reservoir Nodes

Node Number	Name	Dead Storage	Dead Stor Units	Lower Rule	Upper Rule	Max Storage	Max Stor Units
010	Lake Summit (Tuxedo Dam)	1997.6	FT	Pattern	Pattern	2012.6	FT
040	Lake Adger (Turner Shoals Dam)	911.5	FT	Pattern	Pattern	911.6	FT
070	Lake Lure	990.5	FT	Pattern	Pattern	996.4	FT
095	Upper Broad Routing	0.0	AF	None	None	999.0	KAF
220	Cliffside Steam Plant Dam	655.0	FT	Pattern	Pattern	664.0	FT
440	Stice Shoals Dam	654.2	FT	Pattern	Pattern	654.3	FT
525	Lower Broad Routing	0.0	AF	None	None	999.0	KAF
550	Gaston Shoals Dam	570.0	FT	Pattern	Pattern	605.2	FT
600	Kings Mt. / Moss Lake	665.0	FT	Pattern	Pattern	736.0	FT

Reservoir Rules

Node Number	Name	Units	Month	Day	Upper Rule	Lower Rule
010	Lake Summit (Tuxedo Dam)	FT	1	1	2010.60	2005.60
010	Lake Summit (Tuxedo Dam)	FT	12	31	2010.60	2005.60
040	Lake Adger (Turner Shoals Dam)	FT	1	1	911.60	911.50
040	Lake Adger (Turner Shoals Dam)	FT	12	31	911.60	911.50
070	Lake Lure	FT	1	1	996.40	990.50
070	Lake Lure	FT	12	31	996.40	990.50
220	Cliffside Steam Plant Dam	FT	1	1	664.00	655.00
220	Cliffside Steam Plant Dam	FT	12	31	664.00	655.00
440	Stice Shoals Dam	FT	1	1	654.28	654.18
440	Stice Shoals Dam	FT	12	31	654.28	654.18
550	Gaston Shoals Dam	FT	1	1	604.20	603.20
550	Gaston Shoals Dam	FT	2	28	604.20	603.20
550	Gaston Shoals Dam	FT	3	1	605.20	604.20
550	Gaston Shoals Dam	FT	5	31	605.20	604.20
550	Gaston Shoals Dam	FT	6	1	604.20	603.20
550	Gaston Shoals Dam	FT	12	31	604.20	603.20
600	Kings Mt. / Moss Lake	FT	1	1	736.00	718.00
600	Kings Mt. / Moss Lake	FT	12	31	736.00	718.00

Reservoir SAE Data

Node Number	Name	Elevation	Elevation Units	Storage	Storage Units	Area	Area Units
010	Lake Summit (Tuxedo Dam)	1920	FT	0	AF	2	acre
010	Lake Summit (Tuxedo Dam)	1930	FT	50	AF	5	acre
010	Lake Summit (Tuxedo Dam)	1935	FT	100	AF	8	acre
010	Lake Summit (Tuxedo Dam)	1940	FT	150	AF	13	acre
010	Lake Summit (Tuxedo Dam)	1945	FT	250	AF	20	acre
010	Lake Summit (Tuxedo Dam)	1950	FT	395	AF	30	acre
010	Lake Summit (Tuxedo Dam)	1960	FT	875	AF	56	acre
010	Lake Summit (Tuxedo Dam)	1970	FT	1750	AF	88	acre
010	Lake Summit (Tuxedo Dam)	1980	FT	3000	AF	123	acre
010	Lake Summit (Tuxedo Dam)	1990	FT	4550	AF	156	acre
010	Lake Summit (Tuxedo Dam)	2000	FT	6400	AF	200	acre
010	Lake Summit (Tuxedo Dam)	2005	FT	7400	AF	226	acre
010	Lake Summit (Tuxedo Dam)	2010	FT	8650	AF	264	acre
010	Lake Summit (Tuxedo Dam)	2012.6	FT	9343	AF	290	acre
010	Lake Summit (Tuxedo Dam)	2025	FT	15700	AF	387	acre
010	Lake Summit (Tuxedo Dam)	2036.6	FT	22050	AF	438	acre
040	Lake Adger (Turner Shoals Dam)	840	FT	0	AF	5	acre
040	Lake Adger (Turner Shoals Dam)	850	FT	200	AF	22	acre
040	Lake Adger (Turner Shoals Dam)	860	FT	600	AF	52	acre
040	Lake Adger (Turner Shoals Dam)	870	FT	1300	AF	95	acre
040	Lake Adger (Turner Shoals Dam)	880	FT	2500	AF	155	acre
040	Lake Adger (Turner Shoals Dam)	890	FT	4400	AF	225	acre
040	Lake Adger (Turner Shoals Dam)	900	FT	7100	AF	312	acre
040	Lake Adger (Turner Shoals Dam)	910	FT	10700	AF	415	acre
040	Lake Adger (Turner Shoals Dam)	912	FT	11700	AF	438	acre
040	Lake Adger (Turner Shoals Dam)	960	FT	41950	AF	1420	acre
070	Lake Lure	867	FT	0	AF	0	acre
070	Lake Lure	895	FT	798	AF	57	acre
070	Lake Lure	980.4	FT	31840.9	AF	670	acre
070	Lake Lure	991	FT	39345.7	AF	746	acre
070	Lake Lure	1006.4	FT	52096.9	AF	910	acre
070	Lake Lure	1041	FT	89983.6	AF	1280	acre
220	Cliffside Steam Plant Dam	655	FT	0	AF	8.1	acre
220	Cliffside Steam Plant Dam	660	FT	41.5	AF	8.5	acre
220	Cliffside Steam Plant Dam	665	FT	85.4	AF	9	acre
220	Cliffside Steam Plant Dam	670	FT	131.7	AF	9.5	acre
220	Cliffside Steam Plant Dam	675	FT	180.4	AF	10	acre
220	Cliffside Steam Plant Dam	680	FT	231.4	AF	10.4	acre
220	Cliffside Steam Plant Dam	685	FT	284.8	AF	10.9	acre
220	Cliffside Steam Plant Dam	690	FT	340.6	AF	11.4	acre
220	Cliffside Steam Plant Dam	695	FT	398.7	AF	11.9	acre
440	Stice Shoals Dam	635	FT	0	AF	1.8	acre
440	Stice Shoals Dam	640	FT	9	AF	2	acre
440	Stice Shoals Dam	645	FT	20	AF	2.1	acre
440	Stice Shoals Dam	650	FT	31	AF	2.3	acre
440	Stice Shoals Dam	654	FT	41	AF	2.4	acre
440	Stice Shoals Dam	655	FT	42	AF	2.4	acre
440	Stice Shoals Dam	660	FT	55	AF	2.6	acre
440	Stice Shoals Dam	665	FT	68	AF	2.7	acre
440	Stice Shoals Dam	670	FT	82	AF	2.9	acre
440	Stice Shoals Dam	675	FT	97	AF	3	acre
550	Gaston Shoals Dam	560.2	FT	0	AF	0	acre
550	Gaston Shoals Dam	571.5	FT	118.1	AF	20.9	acre

Node Number	Name	Elevation	Elevation Units	Storage	Storage Units	Area	Area Units
550	Gaston Shoals Dam	583	FT	718.4	AF	83.5	acre
550	Gaston Shoals Dam	594	FT	2211.1	AF	187.9	acre
550	Gaston Shoals Dam	605.2	FT	5133.7	AF	334	acre
550	Gaston Shoals Dam	610	FT	7180.9	AF	519	acre
550	Gaston Shoals Dam	615	FT	10343.4	AF	746	acre
550	Gaston Shoals Dam	620	FT	14480.9	AF	909	acre
550	Gaston Shoals Dam	625	FT	19253.4	AF	1000	acre
550	Gaston Shoals Dam	630	FT	24408.4	AF	1062	acre
550	Gaston Shoals Dam	635	FT	29840.9	AF	1111	acre
550	Gaston Shoals Dam	640	FT	35525.9	AF	1163	acre
600	Kings Mt. / Moss Lake	670	FT	0	AF	0	acre
600	Kings Mt. / Moss Lake	680	FT	557	AF	167	acre
600	Kings Mt. / Moss Lake	700	FT	7101	AF	520	acre
600	Kings Mt. / Moss Lake	720	FT	21678	AF	960	acre
600	Kings Mt. / Moss Lake	736	FT	39918	AF	1330	acre
600	Kings Mt. / Moss Lake	740	FT	45553	AF	1489	acre

Demand Patterns

Node Number	Name	Units	Ann. Avg. Dem.	Month	Day	Monthly Factor
061	Polk County	MGD	0	1	1	0.683
061	Polk County	MGD	0	1	31	0.683
061	Polk County	MGD	0	2	1	0.935
061	Polk County	MGD	0	2	29	0.935
061	Polk County	MGD	0	3	1	0.875
061	Polk County	MGD	0	3	31	0.875
061	Polk County	MGD	0	4	1	0.939
061	Polk County	MGD	0	4	30	0.939
061	Polk County	MGD	0	5	1	1.067
061	Polk County	MGD	0	5	31	1.067
061	Polk County	MGD	0	6	1	1.080
061	Polk County	MGD	0	6	30	1.080
061	Polk County	MGD	0	7	1	1.131
061	Polk County	MGD	0	7	31	1.131
061	Polk County	MGD	0	8	1	1.259
061	Polk County	MGD	0	8	31	1.259
061	Polk County	MGD	0	9	1	1.161
061	Polk County	MGD	0	9	30	1.161
061	Polk County	MGD	0	10	1	1.144
061	Polk County	MGD	0	10	31	1.144
061	Polk County	MGD	0	11	1	0.858
061	Polk County	MGD	0	11	30	0.858
061	Polk County	MGD	0	12	1	0.871
061	Polk County	MGD	0	12	31	0.871
086	Broad River Water Authority	MGD	5.17	1	1	0.983
086	Broad River Water Authority	MGD	5.17	1	31	0.983
086	Broad River Water Authority	MGD	5.17	2	1	0.942
086	Broad River Water Authority	MGD	5.17	2	29	0.942
086	Broad River Water Authority	MGD	5.17	3	1	0.942
086	Broad River Water Authority	MGD	5.17	3	31	0.942
086	Broad River Water Authority	MGD	5.17	4	1	0.970
086	Broad River Water Authority	MGD	5.17	4	30	0.970
086	Broad River Water Authority	MGD	5.17	5	1	1.021
086	Broad River Water Authority	MGD	5.17	5	31	1.021
086	Broad River Water Authority	MGD	5.17	6	1	1.081
086	Broad River Water Authority	MGD	5.17	6	30	1.081
086	Broad River Water Authority	MGD	5.17	7	1	1.069
086	Broad River Water Authority	MGD	5.17	7	31	1.069
086	Broad River Water Authority	MGD	5.17	8	1	1.113
086	Broad River Water Authority	MGD	5.17	8	31	1.113
086	Broad River Water Authority	MGD	5.17	9	1	1.047
086	Broad River Water Authority	MGD	5.17	9	30	1.047
086	Broad River Water Authority	MGD	5.17	10	1	1.007
086	Broad River Water Authority	MGD	5.17	10	31	1.007
086	Broad River Water Authority	MGD	5.17	11	1	0.929
086	Broad River Water Authority	MGD	5.17	11	30	0.929
086	Broad River Water Authority	MGD	5.17	12	1	0.896
086	Broad River Water Authority	MGD	5.17	12	31	0.896
186	Forest City	MGD	1.825	1	1	0.992
186	Forest City	MGD	1.825	1	31	0.992
186	Forest City	MGD	1.825	2	1	0.998
186	Forest City	MGD	1.825	2	29	0.998
186	Forest City	MGD	1.825	3	1	0.953

Node Number	Name	Units	Ann. Avg. Dem.	Month	Day	Monthly Factor
186	Forest City	MGD	1.825	3	31	0.953
186	Forest City	MGD	1.825	4	1	1.015
186	Forest City	MGD	1.825	4	30	1.015
186	Forest City	MGD	1.825	5	1	1.033
186	Forest City	MGD	1.825	5	31	1.033
186	Forest City	MGD	1.825	6	1	1.031
186	Forest City	MGD	1.825	6	30	1.031
186	Forest City	MGD	1.825	7	1	0.995
186	Forest City	MGD	1.825	7	31	0.995
186	Forest City	MGD	1.825	8	1	1.042
186	Forest City	MGD	1.825	8	31	1.042
186	Forest City	MGD	1.825	9	1	1.016
186	Forest City	MGD	1.825	9	30	1.016
186	Forest City	MGD	1.825	10	1	1.004
186	Forest City	MGD	1.825	10	31	1.004
186	Forest City	MGD	1.825	11	1	1.001
186	Forest City	MGD	1.825	11	30	1.001
186	Forest City	MGD	1.825	12	1	0.921
186	Forest City	MGD	1.825	12	31	0.921
224	Duke_Cliffside	MGD	19.36	1	1	1.019
224	Duke_Cliffside	MGD	19.36	1	31	1.019
224	Duke_Cliffside	MGD	19.36	2	1	0.995
224	Duke_Cliffside	MGD	19.36	2	29	0.995
224	Duke_Cliffside	MGD	19.36	3	1	0.889
224	Duke_Cliffside	MGD	19.36	3	31	0.889
224	Duke_Cliffside	MGD	19.36	4	1	0.908
224	Duke_Cliffside	MGD	19.36	4	30	0.908
224	Duke_Cliffside	MGD	19.36	5	1	0.964
224	Duke_Cliffside	MGD	19.36	5	31	0.964
224	Duke_Cliffside	MGD	19.36	6	1	1.039
224	Duke_Cliffside	MGD	19.36	6	30	1.039
224	Duke_Cliffside	MGD	19.36	7	1	1.025
224	Duke_Cliffside	MGD	19.36	7	31	1.025
224	Duke_Cliffside	MGD	19.36	8	1	1.084
224	Duke_Cliffside	MGD	19.36	8	31	1.084
224	Duke_Cliffside	MGD	19.36	9	1	1.069
224	Duke_Cliffside	MGD	19.36	9	30	1.069
224	Duke_Cliffside	MGD	19.36	10	1	0.950
224	Duke_Cliffside	MGD	19.36	10	31	0.950
224	Duke_Cliffside	MGD	19.36	11	1	1.046
224	Duke_Cliffside	MGD	19.36	11	30	1.046
224	Duke_Cliffside	MGD	19.36	12	1	1.011
224	Duke_Cliffside	MGD	19.36	12	31	1.011
416	Cleveland Co.	MGD	4.227	1	1	0.902
416	Cleveland Co.	MGD	4.227	1	31	0.902
416	Cleveland Co.	MGD	4.227	2	1	0.893
416	Cleveland Co.	MGD	4.227	2	29	0.893
416	Cleveland Co.	MGD	4.227	3	1	0.902
416	Cleveland Co.	MGD	4.227	3	31	0.902
416	Cleveland Co.	MGD	4.227	4	1	0.959
416	Cleveland Co.	MGD	4.227	4	30	0.959
416	Cleveland Co.	MGD	4.227	5	1	1.046
416	Cleveland Co.	MGD	4.227	5	31	1.046
416	Cleveland Co.	MGD	4.227	6	1	1.135
416	Cleveland Co.	MGD	4.227	6	30	1.135

Node Number	Name	Units	Ann. Avg. Dem.	Month	Day	Monthly Factor
416	Cleveland Co.	MGD	4.227	7	1	1.109
416	Cleveland Co.	MGD	4.227	7	31	1.109
416	Cleveland Co.	MGD	4.227	8	1	1.093
416	Cleveland Co.	MGD	4.227	8	31	1.093
416	Cleveland Co.	MGD	4.227	9	1	1.059
416	Cleveland Co.	MGD	4.227	9	30	1.059
416	Cleveland Co.	MGD	4.227	10	1	0.991
416	Cleveland Co.	MGD	4.227	10	31	0.991
416	Cleveland Co.	MGD	4.227	11	1	0.954
416	Cleveland Co.	MGD	4.227	11	30	0.954
416	Cleveland Co.	MGD	4.227	12	1	0.957
416	Cleveland Co.	MGD	4.227	12	31	0.957
436	Shelby	MGD	3.846	1	1	0.956
436	Shelby	MGD	3.846	1	31	0.956
436	Shelby	MGD	3.846	2	1	0.959
436	Shelby	MGD	3.846	2	29	0.959
436	Shelby	MGD	3.846	3	1	0.946
436	Shelby	MGD	3.846	3	31	0.946
436	Shelby	MGD	3.846	4	1	0.982
436	Shelby	MGD	3.846	4	30	0.982
436	Shelby	MGD	3.846	5	1	1.052
436	Shelby	MGD	3.846	5	31	1.052
436	Shelby	MGD	3.846	6	1	1.159
436	Shelby	MGD	3.846	6	30	1.159
436	Shelby	MGD	3.846	7	1	1.133
436	Shelby	MGD	3.846	7	31	1.133
436	Shelby	MGD	3.846	8	1	1.022
436	Shelby	MGD	3.846	8	31	1.022
436	Shelby	MGD	3.846	9	1	0.987
436	Shelby	MGD	3.846	9	30	0.987
436	Shelby	MGD	3.846	10	1	0.962
436	Shelby	MGD	3.846	10	31	0.962
436	Shelby	MGD	3.846	11	1	0.926
436	Shelby	MGD	3.846	11	30	0.926
436	Shelby	MGD	3.846	12	1	0.917
436	Shelby	MGD	3.846	12	31	0.917
554	Gaffney WS	MGD	8.3	1	1	1.000
554	Gaffney WS	MGD	8.3	12	31	1.000
604	Kings Mt.	MGD	2.43	1	1	0.935
604	Kings Mt.	MGD	2.43	1	31	0.935
604	Kings Mt.	MGD	2.43	2	1	0.969
604	Kings Mt.	MGD	2.43	2	29	0.969
604	Kings Mt.	MGD	2.43	3	1	0.936
604	Kings Mt.	MGD	2.43	3	31	0.936
604	Kings Mt.	MGD	2.43	4	1	0.972
604	Kings Mt.	MGD	2.43	4	30	0.972
604	Kings Mt.	MGD	2.43	5	1	1.034
604	Kings Mt.	MGD	2.43	5	31	1.034
604	Kings Mt.	MGD	2.43	6	1	1.101
604	Kings Mt.	MGD	2.43	6	30	1.101
604	Kings Mt.	MGD	2.43	7	1	1.058
604	Kings Mt.	MGD	2.43	7	31	1.058
604	Kings Mt.	MGD	2.43	8	1	1.157
604	Kings Mt.	MGD	2.43	8	31	1.157
604	Kings Mt.	MGD	2.43	9	1	1.053

Node Number	Name	Units	Ann. Avg. Dem.	Month	Day	Monthly Factor
604	Kings Mt.	MGD	2.43	9	30	1.053
604	Kings Mt.	MGD	2.43	10	1	1.020
604	Kings Mt.	MGD	2.43	10	31	1.020
604	Kings Mt.	MGD	2.43	11	1	0.934
604	Kings Mt.	MGD	2.43	11	30	0.934
604	Kings Mt.	MGD	2.43	12	1	0.831
604	Kings Mt.	MGD	2.43	12	31	0.831
645	CNA Holdings	MGD	0.5	1	1	1.060
645	CNA Holdings	MGD	0.5	1	31	1.060
645	CNA Holdings	MGD	0.5	2	1	0.921
645	CNA Holdings	MGD	0.5	2	28	0.921
645	CNA Holdings	MGD	0.5	3	1	0.809
645	CNA Holdings	MGD	0.5	3	31	0.809
645	CNA Holdings	MGD	0.5	4	1	0.753
645	CNA Holdings	MGD	0.5	4	30	0.753
645	CNA Holdings	MGD	0.5	5	1	0.977
645	CNA Holdings	MGD	0.5	5	31	0.977
645	CNA Holdings	MGD	0.5	6	1	1.005
645	CNA Holdings	MGD	0.5	6	30	1.005
645	CNA Holdings	MGD	0.5	7	1	0.865
645	CNA Holdings	MGD	0.5	7	31	0.865
645	CNA Holdings	MGD	0.5	8	1	0.642
645	CNA Holdings	MGD	0.5	8	31	0.642
645	CNA Holdings	MGD	0.5	9	1	0.977
645	CNA Holdings	MGD	0.5	9	30	0.977
645	CNA Holdings	MGD	0.5	10	1	1.228
645	CNA Holdings	MGD	0.5	10	31	1.228
645	CNA Holdings	MGD	0.5	11	1	1.451
645	CNA Holdings	MGD	0.5	11	30	1.451
645	CNA Holdings	MGD	0.5	12	1	1.312
645	CNA Holdings	MGD	0.5	12	31	1.312

Wastewater Return Patterns (as fraction of demand)

Name	Month	Factor
BSpringsWW	1	0.062
BSpringsWW	2	0.061
BSpringsWW	3	0.064
BSpringsWW	4	0.058
BSpringsWW	5	0.044
BSpringsWW	6	0.037
BSpringsWW	7	0.04
BSpringsWW	8	0.048
BSpringsWW	9	0.059
BSpringsWW	10	0.054
BSpringsWW	11	0.058
BSpringsWW	12	0.061
CliffsideWW	1	0.399
CliffsideWW	2	0.418
CliffsideWW	3	0.491
CliffsideWW	4	0.44
CliffsideWW	5	0.421
CliffsideWW	6	0.437
CliffsideWW	7	0.443
CliffsideWW	8	0.444
CliffsideWW	9	0.403
CliffsideWW	10	0.419
CliffsideWW	11	0.403
CliffsideWW	12	0.447
CNAWW	1	0.895
CNAWW	2	0.895
CNAWW	3	0.895
CNAWW	4	0.895
CNAWW	5	0.895
CNAWW	6	0.895
CNAWW	7	0.895
CNAWW	8	0.895
CNAWW	9	0.895
CNAWW	10	0.895
CNAWW	11	0.895
CNAWW	12	0.895
ForestCityWW	1	0.13

Name	Month	Factor
KingsMtWW	1	0.982
KingsMtWW	2	0.868
KingsMtWW	3	0.996
KingsMtWW	4	0.845
KingsMtWW	5	0.796
KingsMtWW	6	0.704
KingsMtWW	7	0.631
KingsMtWW	8	0.639
KingsMtWW	9	0.753
KingsMtWW	10	0.743
KingsMtWW	11	0.827
KingsMtWW	12	0.946
RuthefordWW	1	0.132
RuthefordWW	2	0.135
RuthefordWW	3	0.143
RuthefordWW	4	0.127
RuthefordWW	5	0.121
RuthefordWW	6	0.115
RuthefordWW	7	0.121
RuthefordWW	8	0.112
RuthefordWW	9	0.127
RuthefordWW	10	0.125
RuthefordWW	11	0.14
RuthefordWW	12	0.165
ShelbyWW	1	0.626
ShelbyWW	2	0.593
ShelbyWW	3	0.681
ShelbyWW	4	0.569
ShelbyWW	5	0.497
ShelbyWW	6	0.416
ShelbyWW	7	0.417
ShelbyWW	8	0.429
ShelbyWW	9	0.475
ShelbyWW	10	0.477
ShelbyWW	11	0.509
ShelbyWW	12	0.601
SpindaleWW	1	0.3

Name	Month	Factor
ForestCityWW	2	0.705
ForestCityWW	3	0.698
ForestCityWW	4	0.758
ForestCityWW	5	0.702
ForestCityWW	6	0.65
ForestCityWW	7	0.688
ForestCityWW	8	0.677
ForestCityWW	9	0.696
ForestCityWW	10	0.686
ForestCityWW	11	0.643
ForestCityWW	12	0.65
GroverWW	1	0.003
GroverWW	2	0.002
GroverWW	3	0.002
GroverWW	4	0.002
GroverWW	5	0.003
GroverWW	6	0.001
GroverWW	7	0.001
GroverWW	8	0.002
GroverWW	9	0.002
GroverWW	10	0.001
GroverWW	11	0.002
GroverWW	12	0.002

Name	Month	Factor
SpindaleWW	2	0.291
SpindaleWW	3	0.297
SpindaleWW	4	0.268
SpindaleWW	5	0.266
SpindaleWW	6	0.286
SpindaleWW	7	0.264
SpindaleWW	8	0.274
SpindaleWW	9	0.297
SpindaleWW	10	0.27
SpindaleWW	11	0.301
SpindaleWW	12	0.316

Arc Minimum Flows

Name	U/S Node	D/S Node	Units	Month	Day	Min Flow
Lake Lure	070	080	CFS	1	1	7
Lake Lure	070	080	CFS	12	31	7
1st Broad Shelby	420	440	CFS	1	1	25
1st Broad Shelby	420	440	CFS	12	31	25
Gaston Shoals Bypass	550	555	CFS	1	1	150
Gaston Shoals Bypass	550	555	CFS	2	28	150
Gaston Shoals Bypass	550	555	CFS	3	1	350
Gaston Shoals Bypass	550	555	CFS	5	31	350
Gaston Shoals Bypass	550	555	CFS	6	1	150
Gaston Shoals Bypass	550	555	CFS	12	31	150
Gaston Shoals Release	550	700	CFS	1	1	718
Gaston Shoals Release	550	700	CFS	2	29	718
Gaston Shoals Release	550	700	CFS	3	1	518
Gaston Shoals Release	550	700	CFS	4	30	518
Gaston Shoals Release	550	700	CFS	5	1	301
Gaston Shoals Release	550	700	CFS	5	31	301
Gaston Shoals Release	550	700	CFS	6	1	501
Gaston Shoals Release	550	700	CFS	6	30	501
Gaston Shoals Release	550	700	CFS	7	1	284
Gaston Shoals Release	550	700	CFS	11	30	284
Gaston Shoals Release	550	700	CFS	12	1	501
Gaston Shoals Release	550	700	CFS	12	31	501
Moss Lake	600	610	CFS	1	1	12
Moss Lake	600	610	CFS	12	31	12

County Irrigation Data

Crop	Units	Bunc	Clevd	Gast	Hend	Linc	McDo	Polk	Ruth	Cher_SC	Spart_SC
IrrTobacco	ac.	0	0	0	0	0	0	0	0	0	0
Turf	ac.	0	6.825	0.66	45	0	0	0	2.764	0	0.075
Golf	ac.	0	0	0	24.594	0	0	92.964	0.558	0	0
ContNurs	ac.	0	0	0.025	0.689	5.186	0	0	0	0	0
FieldNurs	ac.	0	0	0.147	4.552	20.743	0	0	0	0	0.652
IrrCotton	ac.	0	0	0	0	0	0	0	0	0	0
IrrEarlySoy	ac.	0	0	0	0	0	0	0	0	0	0.018
IrrLateSoy	ac.	0	0	0	0	0	0	0	0	0	0.018
IrrCorn	ac.	0	0	3.3	0	0	0	0	0	6.534	0
IrrVeg	ac.	2.5	5.13	0.106	0	112.5	0	64.38	0	9.348	0.064
IrrPas&Hay	ac.	0	0	0.825	0	0	0	2.688	0	0	0.963
IrrPeanut	ac.	0	0	0	0	0	0	0	0	0	0
IrrBlueberry	ac.	0	0	0	0	0	0	0	0	0.36	0.008
IrrStrawberry	ac.	0	0	0	0	0	0	0	0	0	0
IrrFruit	ac.	0	48.983	0	0	47.025	0	0	0	0.42	0.98
Beef Cattle	#	76	1452	14	149.6	0	0	432.4	64	435.6	34.8
Dairy Cows	#	0	76.23	6	102	0	0	271.14	1.49	0	4.088
Horses	#	15.807	157.687	7.265	78.268	0	0	112.9	70.019	79.774	9.262
Pigs	#	0	84.942	0.424	5.372	0	0	0	2.787	1.723	0.099
Chickens	#	0	0	1031.773	148.58	0	0	0	3360	0	9.662
Turkeys	#	0	0	0.234	2.992	0	0	0	1.78	0	560
Other Animals	#	13.775	135.689	7.305	69.836	0	0	104.841	91.817	94.723	9.945

Crop Irrigation Coefficients

Name	Month	Day	Value
IrrCoef_Beef	1	1	12
IrrCoef_Beef	12	31	12
IrrCoef_Blueberry	1	1	0
IrrCoef_Blueberry	2	28	0
IrrCoef_Blueberry	3	1	1
IrrCoef_Blueberry	4	14	1
IrrCoef_Blueberry	4	15	0.178571
IrrCoef_Blueberry	9	30	0.178571
IrrCoef_Blueberry	10	1	0
IrrCoef_Blueberry	12	31	0
IrrCoef_Chicken	1	1	9
IrrCoef_Chicken	12	31	9
IrrCoef_ContNurs	1	1	0.2
IrrCoef_ContNurs	3	31	0.2
IrrCoef_ContNurs	4	1	0.5
IrrCoef_ContNurs	5	31	0.5
IrrCoef_ContNurs	6	1	0.75
IrrCoef_ContNurs	8	31	0.75
IrrCoef_ContNurs	9	1	0.5
IrrCoef_ContNurs	10	31	0.5
IrrCoef_ContNurs	11	1	0.2
IrrCoef_ContNurs	12	31	0.2
IrrCoef_Corn	1	1	0
IrrCoef_Corn	4	23	0
IrrCoef_Corn	5	4	0.02
IrrCoef_Corn	5	14	0.039
IrrCoef_Corn	5	23	0.0543
IrrCoef_Corn	5	24	0.056
IrrCoef_Corn	6	2	0.0722
IrrCoef_Corn	6	3	0.074
IrrCoef_Corn	6	12	0.0974
IrrCoef_Corn	6	13	0.1
IrrCoef_Corn	6	18	0.115
IrrCoef_Corn	6	23	0.1425
IrrCoef_Corn	7	2	0.21675
IrrCoef_Corn	7	3	0.225
IrrCoef_Corn	7	4	0.229
IrrCoef_Corn	7	5	0.233
IrrCoef_Corn	7	6	0.237
IrrCoef_Corn	7	7	0.241
IrrCoef_Corn	7	8	0.245
IrrCoef_Corn	7	9	0.247
IrrCoef_Corn	7	10	0.249
IrrCoef_Corn	7	11	0.251
IrrCoef_Corn	7	12	0.253
IrrCoef_Corn	7	13	0.255
IrrCoef_Corn	7	17	0.261
IrrCoef_Corn	7	18	0.2625
IrrCoef_Corn	7	22	0.2565
IrrCoef_Corn	7	23	0.255
IrrCoef_Corn	8	1	0.23925
IrrCoef_Corn	8	2	0.2375
IrrCoef_Corn	8	11	0.206
IrrCoef_Corn	8	12	0.2025

Name	Month	Day	Value
IrrCoef_OtherAnimal	1	1	2
IrrCoef_OtherAnimal	12	31	2
IrrCoef_PastHay	1	1	0
IrrCoef_PastHay	4	30	0
IrrCoef_PastHay	5	1	0.142857
IrrCoef_PastHay	9	30	0.142857
IrrCoef_PastHay	10	1	0
IrrCoef_PastHay	12	31	0
IrrCoef_Peanut	1	1	0
IrrCoef_Peanut	5	7	0
IrrCoef_Peanut	5	8	0.00014
IrrCoef_Peanut	5	9	0.00028
IrrCoef_Peanut	5	10	0.00042
IrrCoef_Peanut	5	11	0.00056
IrrCoef_Peanut	5	12	0.0007
IrrCoef_Peanut	5	13	0.00084
IrrCoef_Peanut	5	14	0.00098
IrrCoef_Peanut	5	15	0.00112
IrrCoef_Peanut	5	16	0.00126
IrrCoef_Peanut	5	17	0.0014
IrrCoef_Peanut	5	18	0.0028
IrrCoef_Peanut	5	19	0.0042
IrrCoef_Peanut	5	20	0.0056
IrrCoef_Peanut	5	21	0.007
IrrCoef_Peanut	5	22	0.0084
IrrCoef_Peanut	5	23	0.0098
IrrCoef_Peanut	5	24	0.0112
IrrCoef_Peanut	5	25	0.0126
IrrCoef_Peanut	5	26	0.014
IrrCoef_Peanut	5	27	0.0158
IrrCoef_Peanut	5	28	0.0176
IrrCoef_Peanut	5	29	0.0194
IrrCoef_Peanut	5	30	0.0212
IrrCoef_Peanut	5	31	0.023
IrrCoef_Peanut	6	1	0.0248
IrrCoef_Peanut	6	2	0.0266
IrrCoef_Peanut	6	3	0.0284
IrrCoef_Peanut	6	4	0.0302
IrrCoef_Peanut	6	5	0.032
IrrCoef_Peanut	6	6	0.0338
IrrCoef_Peanut	6	7	0.0356
IrrCoef_Peanut	6	8	0.0374
IrrCoef_Peanut	6	9	0.0392
IrrCoef_Peanut	6	10	0.041
IrrCoef_Peanut	6	11	0.0428
IrrCoef_Peanut	6	12	0.0446
IrrCoef_Peanut	6	13	0.0464
IrrCoef_Peanut	6	14	0.0482
IrrCoef_Peanut	6	15	0.05
IrrCoef_Peanut	6	16	0.0526
IrrCoef_Peanut	6	17	0.0552
IrrCoef_Peanut	6	18	0.0578
IrrCoef_Peanut	6	19	0.0604
IrrCoef_Peanut	6	20	0.063

Name	Month	Day	Value
IrrCoef_Corn	8	21	0.1665
IrrCoef_Corn	8	22	0.1625
IrrCoef_Corn	8	31	0.12875
IrrCoef_Corn	9	1	0.125
IrrCoef_Corn	9	10	0.0935
IrrCoef_Corn	9	11	0.09
IrrCoef_Corn	9	20	0.072
IrrCoef_Corn	9	21	0.07
IrrCoef_Corn	9	30	0.061
IrrCoef_Corn	10	1	0.06
IrrCoef_Corn	10	12	0.050833
IrrCoef_Corn	10	13	0.05
IrrCoef_Corn	10	14	0
IrrCoef_Corn	12	31	0
IrrCoef_Cotton	1	1	0
IrrCoef_Cotton	5	7	0
IrrCoef_Cotton	5	8	0.001
IrrCoef_Cotton	5	9	0.002
IrrCoef_Cotton	5	30	0.0288
IrrCoef_Cotton	5	31	0.0304
IrrCoef_Cotton	6	1	0.032
IrrCoef_Cotton	6	2	0.0336
IrrCoef_Cotton	6	3	0.0352
IrrCoef_Cotton	6	4	0.0368
IrrCoef_Cotton	6	5	0.0384
IrrCoef_Cotton	6	6	0.04
IrrCoef_Cotton	6	7	0.0417
IrrCoef_Cotton	6	8	0.0434
IrrCoef_Cotton	6	9	0.0451
IrrCoef_Cotton	6	10	0.0468
IrrCoef_Cotton	6	11	0.0485
IrrCoef_Cotton	6	12	0.0502
IrrCoef_Cotton	6	13	0.0519
IrrCoef_Cotton	6	14	0.0536
IrrCoef_Cotton	6	15	0.0553
IrrCoef_Cotton	6	16	0.057
IrrCoef_Cotton	6	17	0.0587
IrrCoef_Cotton	6	18	0.0604
IrrCoef_Cotton	6	19	0.0621
IrrCoef_Cotton	6	20	0.0638
IrrCoef_Cotton	6	21	0.0655
IrrCoef_Cotton	6	22	0.0672
IrrCoef_Cotton	6	23	0.0689
IrrCoef_Cotton	6	24	0.0706
IrrCoef_Cotton	6	25	0.0723
IrrCoef_Cotton	6	26	0.074
IrrCoef_Cotton	6	27	0.0764
IrrCoef_Cotton	6	28	0.0788
IrrCoef_Cotton	6	29	0.0812
IrrCoef_Cotton	6	30	0.0836
IrrCoef_Cotton	7	1	0.086
IrrCoef_Cotton	7	2	0.0884
IrrCoef_Cotton	7	3	0.0908
IrrCoef_Cotton	7	4	0.0932
IrrCoef_Cotton	7	5	0.0956
IrrCoef_Cotton	7	6	0.098

Name	Month	Day	Value
IrrCoef_Peanut	6	21	0.0656
IrrCoef_Peanut	6	22	0.0682
IrrCoef_Peanut	6	23	0.0708
IrrCoef_Peanut	6	24	0.0734
IrrCoef_Peanut	6	25	0.076
IrrCoef_Peanut	6	26	0.07915
IrrCoef_Peanut	6	27	0.0823
IrrCoef_Peanut	6	28	0.08545
IrrCoef_Peanut	6	29	0.0886
IrrCoef_Peanut	6	30	0.09175
IrrCoef_Peanut	7	1	0.0949
IrrCoef_Peanut	7	2	0.09805
IrrCoef_Peanut	7	3	0.1012
IrrCoef_Peanut	7	4	0.10435
IrrCoef_Peanut	7	5	0.1075
IrrCoef_Peanut	7	6	0.113
IrrCoef_Peanut	7	7	0.1185
IrrCoef_Peanut	7	8	0.124
IrrCoef_Peanut	7	9	0.1295
IrrCoef_Peanut	7	10	0.135
IrrCoef_Peanut	7	11	0.1405
IrrCoef_Peanut	7	12	0.146
IrrCoef_Peanut	7	13	0.1515
IrrCoef_Peanut	7	14	0.157
IrrCoef_Peanut	7	15	0.1625
IrrCoef_Peanut	7	16	0.168
IrrCoef_Peanut	7	17	0.1735
IrrCoef_Peanut	7	18	0.179
IrrCoef_Peanut	7	19	0.1845
IrrCoef_Peanut	7	20	0.19
IrrCoef_Peanut	7	21	0.1955
IrrCoef_Peanut	7	22	0.201
IrrCoef_Peanut	7	23	0.2065
IrrCoef_Peanut	7	24	0.212
IrrCoef_Peanut	7	25	0.2175
IrrCoef_Peanut	7	26	0.22
IrrCoef_Peanut	7	27	0.2225
IrrCoef_Peanut	7	28	0.225
IrrCoef_Peanut	7	29	0.2275
IrrCoef_Peanut	7	30	0.23
IrrCoef_Peanut	7	31	0.2325
IrrCoef_Peanut	8	1	0.235
IrrCoef_Peanut	8	2	0.2375
IrrCoef_Peanut	8	3	0.24
IrrCoef_Peanut	8	4	0.2425
IrrCoef_Peanut	8	5	0.243
IrrCoef_Peanut	8	6	0.2435
IrrCoef_Peanut	8	7	0.244
IrrCoef_Peanut	8	8	0.2445
IrrCoef_Peanut	8	9	0.245
IrrCoef_Peanut	8	10	0.2455
IrrCoef_Peanut	8	11	0.246
IrrCoef_Peanut	8	12	0.2465
IrrCoef_Peanut	8	13	0.247
IrrCoef_Peanut	8	14	0.2475
IrrCoef_Peanut	8	15	0.247

Name	Month	Day	Value
IrrCoef_Cotton	7	7	0.1009
IrrCoef_Cotton	7	8	0.1038
IrrCoef_Cotton	7	9	0.1067
IrrCoef_Cotton	7	10	0.1096
IrrCoef_Cotton	7	11	0.1125
IrrCoef_Cotton	7	12	0.1175
IrrCoef_Cotton	7	13	0.1225
IrrCoef_Cotton	7	14	0.1275
IrrCoef_Cotton	7	15	0.1325
IrrCoef_Cotton	7	16	0.1375
IrrCoef_Cotton	7	17	0.145
IrrCoef_Cotton	7	18	0.1525
IrrCoef_Cotton	7	19	0.16
IrrCoef_Cotton	7	20	0.1675
IrrCoef_Cotton	7	21	0.175
IrrCoef_Cotton	7	22	0.1825
IrrCoef_Cotton	7	23	0.19
IrrCoef_Cotton	7	24	0.1975
IrrCoef_Cotton	7	25	0.205
IrrCoef_Cotton	7	26	0.2125
IrrCoef_Cotton	7	27	0.2175
IrrCoef_Cotton	7	28	0.2225
IrrCoef_Cotton	7	29	0.2275
IrrCoef_Cotton	7	30	0.2325
IrrCoef_Cotton	7	31	0.2375
IrrCoef_Cotton	8	1	0.24
IrrCoef_Cotton	8	2	0.2425
IrrCoef_Cotton	8	3	0.245
IrrCoef_Cotton	8	4	0.2475
IrrCoef_Cotton	8	5	0.25
IrrCoef_Cotton	8	6	0.25
IrrCoef_Cotton	8	7	0.25
IrrCoef_Cotton	8	8	0.25
IrrCoef_Cotton	8	9	0.25
IrrCoef_Cotton	8	10	0.25
IrrCoef_Cotton	8	11	0.25
IrrCoef_Cotton	8	12	0.25
IrrCoef_Cotton	8	13	0.25
IrrCoef_Cotton	8	14	0.25
IrrCoef_Cotton	8	15	0.25
IrrCoef_Cotton	8	16	0.2485
IrrCoef_Cotton	8	17	0.247
IrrCoef_Cotton	8	18	0.2455
IrrCoef_Cotton	8	19	0.244
IrrCoef_Cotton	8	20	0.2425
IrrCoef_Cotton	8	21	0.241
IrrCoef_Cotton	8	22	0.2395
IrrCoef_Cotton	8	23	0.238
IrrCoef_Cotton	8	24	0.2365
IrrCoef_Cotton	8	25	0.235
IrrCoef_Cotton	8	26	0.232
IrrCoef_Cotton	8	27	0.229
IrrCoef_Cotton	8	28	0.226
IrrCoef_Cotton	8	29	0.223
IrrCoef_Cotton	8	30	0.22
IrrCoef_Cotton	8	31	0.217

Name	Month	Day	Value
IrrCoef_Peanut	8	16	0.2465
IrrCoef_Peanut	8	17	0.246
IrrCoef_Peanut	8	18	0.2455
IrrCoef_Peanut	8	19	0.245
IrrCoef_Peanut	8	20	0.2445
IrrCoef_Peanut	8	21	0.244
IrrCoef_Peanut	8	22	0.2435
IrrCoef_Peanut	8	23	0.243
IrrCoef_Peanut	8	24	0.2425
IrrCoef_Peanut	8	25	0.24075
IrrCoef_Peanut	8	26	0.239
IrrCoef_Peanut	8	27	0.23725
IrrCoef_Peanut	8	28	0.2355
IrrCoef_Peanut	8	29	0.23375
IrrCoef_Peanut	8	30	0.232
IrrCoef_Peanut	8	31	0.23025
IrrCoef_Peanut	9	1	0.2285
IrrCoef_Peanut	9	2	0.22675
IrrCoef_Peanut	9	3	0.225
IrrCoef_Peanut	9	4	0.22275
IrrCoef_Peanut	9	5	0.2205
IrrCoef_Peanut	9	6	0.21825
IrrCoef_Peanut	9	7	0.216
IrrCoef_Peanut	9	8	0.21375
IrrCoef_Peanut	9	9	0.2115
IrrCoef_Peanut	9	10	0.20925
IrrCoef_Peanut	9	11	0.207
IrrCoef_Peanut	9	12	0.20475
IrrCoef_Peanut	9	13	0.2025
IrrCoef_Peanut	9	14	0.19975
IrrCoef_Peanut	9	15	0.197
IrrCoef_Peanut	9	16	0.19425
IrrCoef_Peanut	9	17	0.1915
IrrCoef_Peanut	9	18	0.18875
IrrCoef_Peanut	9	19	0.186
IrrCoef_Peanut	9	20	0.18325
IrrCoef_Peanut	9	21	0.1805
IrrCoef_Peanut	9	22	0.17775
IrrCoef_Peanut	9	23	0.175
IrrCoef_Peanut	9	24	0.1725
IrrCoef_Peanut	9	25	0.17
IrrCoef_Peanut	9	26	0.1675
IrrCoef_Peanut	9	27	0.165
IrrCoef_Peanut	9	28	0.1625
IrrCoef_Peanut	9	29	0.16
IrrCoef_Peanut	9	30	0.1575
IrrCoef_Peanut	10	1	0.155
IrrCoef_Peanut	10	2	0.1525
IrrCoef_Peanut	10	3	0.15
IrrCoef_Peanut	10	4	0.14775
IrrCoef_Peanut	10	5	0.1455
IrrCoef_Peanut	10	6	0.14325
IrrCoef_Peanut	10	7	0.141
IrrCoef_Peanut	10	8	0.13875
IrrCoef_Peanut	10	9	0.1365
IrrCoef_Peanut	10	10	0.13425

Name	Month	Day	Value
IrrCoef_Cotton	9	1	0.214
IrrCoef_Cotton	9	2	0.211
IrrCoef_Cotton	9	3	0.208
IrrCoef_Cotton	9	4	0.205
IrrCoef_Cotton	9	5	0.202
IrrCoef_Cotton	9	6	0.199
IrrCoef_Cotton	9	7	0.196
IrrCoef_Cotton	9	8	0.193
IrrCoef_Cotton	9	9	0.19
IrrCoef_Cotton	9	10	0.187
IrrCoef_Cotton	9	11	0.184
IrrCoef_Cotton	9	12	0.181
IrrCoef_Cotton	9	13	0.178
IrrCoef_Cotton	9	14	0.175
IrrCoef_Cotton	9	15	0.1725
IrrCoef_Cotton	9	16	0.17
IrrCoef_Cotton	9	17	0.1675
IrrCoef_Cotton	9	18	0.165
IrrCoef_Cotton	9	19	0.1625
IrrCoef_Cotton	9	20	0.16
IrrCoef_Cotton	9	21	0.1575
IrrCoef_Cotton	9	22	0.155
IrrCoef_Cotton	9	23	0.1525
IrrCoef_Cotton	9	24	0.15
IrrCoef_Cotton	9	25	0.1475
IrrCoef_Cotton	9	26	0.145
IrrCoef_Cotton	9	27	0.1425
IrrCoef_Cotton	9	28	0.14
IrrCoef_Cotton	9	29	0.1375
IrrCoef_Cotton	9	30	0.135
IrrCoef_Cotton	10	1	0.1325
IrrCoef_Cotton	10	2	0.13
IrrCoef_Cotton	10	3	0.1275
IrrCoef_Cotton	10	4	0.125
IrrCoef_Cotton	10	5	0.12275
IrrCoef_Cotton	10	6	0.1205
IrrCoef_Cotton	10	7	0.11825
IrrCoef_Cotton	10	8	0.116
IrrCoef_Cotton	10	9	0.11375
IrrCoef_Cotton	10	10	0.1115
IrrCoef_Cotton	10	11	0.10925
IrrCoef_Cotton	10	12	0.107
IrrCoef_Cotton	10	13	0.10475
IrrCoef_Cotton	10	14	0.1025
IrrCoef_Cotton	10	15	0
IrrCoef_Cotton	12	31	0
IrrCoef_Cotton	5	10	0.003
IrrCoef_Cotton	5	11	0.004
IrrCoef_Cotton	5	12	0.005
IrrCoef_Cotton	5	13	0.006
IrrCoef_Cotton	5	14	0.007
IrrCoef_Cotton	5	15	0.008
IrrCoef_Cotton	5	16	0.009
IrrCoef_Cotton	5	17	0.01
IrrCoef_Cotton	5	18	0.0114
IrrCoef_Cotton	5	19	0.0128

Name	Month	Day	Value
IrrCoef_Peanut	10	11	0.132
IrrCoef_Peanut	10	12	0.12975
IrrCoef_Peanut	10	13	0.1275
IrrCoef_Peanut	10	14	0.126
IrrCoef_Peanut	10	15	0.1245
IrrCoef_Peanut	10	16	0.123
IrrCoef_Peanut	10	17	0.1215
IrrCoef_Peanut	10	18	0.12
IrrCoef_Peanut	10	19	0.1185
IrrCoef_Peanut	10	20	0.117
IrrCoef_Peanut	10	21	0.1155
IrrCoef_Peanut	10	22	0.114
IrrCoef_Peanut	10	23	0.1125
IrrCoef_Peanut	10	24	0.110417
IrrCoef_Peanut	10	25	0.108333
IrrCoef_Peanut	10	26	0.10625
IrrCoef_Peanut	10	27	0.104167
IrrCoef_Peanut	10	28	0.102083
IrrCoef_Peanut	10	29	0.1
IrrCoef_Peanut	10	30	0
IrrCoef_Peanut	12	31	0
IrrCoef_Pig	1	1	4
IrrCoef_Pig	12	31	4
IrrCoef_Strawberry	1	1	0
IrrCoef_Strawberry	2	28	0
IrrCoef_Strawberry	3	1	1
IrrCoef_Strawberry	3	31	1
IrrCoef_Strawberry	4	1	0.178571
IrrCoef_Strawberry	5	31	0.178571
IrrCoef_Strawberry	6	1	0
IrrCoef_Strawberry	9	14	0
IrrCoef_Strawberry	9	15	0.178571
IrrCoef_Strawberry	9	30	0.178571
IrrCoef_Strawberry	10	1	1
IrrCoef_Strawberry	11	15	1
IrrCoef_Strawberry	11	16	0
IrrCoef_Strawberry	12	31	0
IrrCoef_Tobacco	1	1	0
IrrCoef_Tobacco	5	20	0
IrrCoef_Tobacco	5	21	0.06
IrrCoef_Tobacco	6	10	0.06
IrrCoef_Tobacco	6	11	0.062
IrrCoef_Tobacco	6	12	0.064
IrrCoef_Tobacco	6	22	0.083333
IrrCoef_Tobacco	6	23	0.086667
IrrCoef_Tobacco	6	24	0.09
IrrCoef_Tobacco	6	25	0.0933
IrrCoef_Tobacco	6	26	0.0967
IrrCoef_Tobacco	6	27	0.1
IrrCoef_Tobacco	6	28	0.10625
IrrCoef_Tobacco	6	29	0.1125
IrrCoef_Tobacco	6	30	0.11875
IrrCoef_Tobacco	7	1	0.125
IrrCoef_Tobacco	7	2	0.133333
IrrCoef_Tobacco	7	3	0.141667
IrrCoef_Tobacco	7	4	0.15

Name	Month	Day	Value
IrrCoef_Cotton	5	20	0.0142
IrrCoef_Cotton	5	21	0.0156
IrrCoef_Cotton	5	22	0.017
IrrCoef_Cotton	5	23	0.0184
IrrCoef_Cotton	5	24	0.0198
IrrCoef_Cotton	5	25	0.0212
IrrCoef_Cotton	5	26	0.0226
IrrCoef_Cotton	5	27	0.024
IrrCoef_Cotton	5	28	0.0256
IrrCoef_Cotton	5	29	0.0272
IrrCoef_Dairy	1	1	40
IrrCoef_Dairy	12	31	40
IrrCoef_EarlySoy	1	1	0
IrrCoef_EarlySoy	5	20	0
IrrCoef_EarlySoy	5	21	0.001
IrrCoef_EarlySoy	5	30	0.01
IrrCoef_EarlySoy	5	31	0.012
IrrCoef_EarlySoy	6	19	0.05
IrrCoef_EarlySoy	6	29	0.075
IrrCoef_EarlySoy	7	9	0.11
IrrCoef_EarlySoy	7	19	0.16
IrrCoef_EarlySoy	7	29	0.2025
IrrCoef_EarlySoy	8	8	0.2375
IrrCoef_EarlySoy	8	18	0.2525
IrrCoef_EarlySoy	8	28	0.2475
IrrCoef_EarlySoy	9	7	0.21
IrrCoef_EarlySoy	9	27	0.09
IrrCoef_EarlySoy	10	7	0.059
IrrCoef_EarlySoy	10	17	0.034
IrrCoef_EarlySoy	10	27	0.019
IrrCoef_EarlySoy	10	28	0
IrrCoef_EarlySoy	12	31	0
IrrCoef_FieldNurs	1	1	0
IrrCoef_FieldNurs	4	30	0
IrrCoef_FieldNurs	5	1	0.178571
IrrCoef_FieldNurs	10	31	0.178571
IrrCoef_FieldNurs	11	1	0
IrrCoef_FieldNurs	12	31	0
IrrCoef_Fruit	1	1	0
IrrCoef_Fruit	2	28	0
IrrCoef_Fruit	3	1	1.214286
IrrCoef_Fruit	4	15	1.214286
IrrCoef_Fruit	4	16	0.178571
IrrCoef_Fruit	8	31	0.178571
IrrCoef_Fruit	9	1	0
IrrCoef_Fruit	12	31	0
IrrCoef_Golf	1	1	0.0145
IrrCoef_Golf	3	31	0.0145
IrrCoef_Golf	4	1	0.081429
IrrCoef_Golf	10	31	0.081429
IrrCoef_Golf	11	1	0.0145
IrrCoef_Golf	12	31	0.0145
IrrCoef_Horse	1	1	12
IrrCoef_Horse	12	31	12
IrrCoef_LateSoy	1	1	0
IrrCoef_LateSoy	6	20	0

Name	Month	Day	Value
IrrCoef_Tobacco	7	5	0.155625
IrrCoef_Tobacco	7	6	0.16125
IrrCoef_Tobacco	7	7	0.166875
IrrCoef_Tobacco	7	8	0.1725
IrrCoef_Tobacco	7	9	0.18
IrrCoef_Tobacco	7	10	0.1875
IrrCoef_Tobacco	7	11	0.195
IrrCoef_Tobacco	7	12	0.200625
IrrCoef_Tobacco	7	13	0.20625
IrrCoef_Tobacco	7	14	0.211875
IrrCoef_Tobacco	7	15	0.2175
IrrCoef_Tobacco	7	18	0.2325
IrrCoef_Tobacco	7	23	0.24
IrrCoef_Tobacco	7	24	0.241667
IrrCoef_Tobacco	7	25	0.243333
IrrCoef_Tobacco	7	26	0.245
IrrCoef_Tobacco	7	30	0.24
IrrCoef_Tobacco	7	31	0.235
IrrCoef_Tobacco	8	1	0.23
IrrCoef_Tobacco	8	2	0.225
IrrCoef_Tobacco	8	3	0.219375
IrrCoef_Tobacco	8	4	0.21375
IrrCoef_Tobacco	8	5	0.208125
IrrCoef_Tobacco	8	6	0.2025
IrrCoef_Tobacco	8	7	0.193333
IrrCoef_Tobacco	8	8	0.184167
IrrCoef_Tobacco	8	9	0.175
IrrCoef_Tobacco	8	10	0.17
IrrCoef_Tobacco	8	11	0.165
IrrCoef_Tobacco	8	12	0.16
IrrCoef_Tobacco	8	13	0.155
IrrCoef_Tobacco	8	14	0.1475
IrrCoef_Tobacco	8	18	0.125
IrrCoef_Tobacco	8	19	0.12125
IrrCoef_Tobacco	8	20	0.1175
IrrCoef_Tobacco	8	21	0.113333
IrrCoef_Tobacco	8	22	0.109167
IrrCoef_Tobacco	8	23	0.105
IrrCoef_Tobacco	8	24	0.10125
IrrCoef_Tobacco	8	25	0.0975
IrrCoef_Tobacco	8	26	0.09375
IrrCoef_Tobacco	8	27	0.09
IrrCoef_Tobacco	8	28	0.089
IrrCoef_Tobacco	8	29	0.088
IrrCoef_Tobacco	9	6	0.08
IrrCoef_Tobacco	9	7	0
IrrCoef_Tobacco	12	31	0
IrrCoef_Turf	1	1	0
IrrCoef_Turf	4	14	0
IrrCoef_Turf	4	15	0.178571
IrrCoef_Turf	10	15	0.178571
IrrCoef_Turf	10	16	0
IrrCoef_Turf	12	31	0
IrrCoef_Turkey	1	1	9
IrrCoef_Turkey	12	31	9
IrrCoef_Veg	1	1	0

Name	Month	Day	Value
IrrCoef_LateSoy	6	21	0.001
IrrCoef_LateSoy	6	30	0.01
IrrCoef_LateSoy	7	1	0.012
IrrCoef_LateSoy	7	20	0.05
IrrCoef_LateSoy	7	30	0.075
IrrCoef_LateSoy	8	9	0.11
IrrCoef_LateSoy	8	19	0.16
IrrCoef_LateSoy	8	29	0.2025
IrrCoef_LateSoy	9	8	0.2375
IrrCoef_LateSoy	9	18	0.2525
IrrCoef_LateSoy	9	28	0.2475
IrrCoef_LateSoy	10	8	0.21
IrrCoef_LateSoy	10	28	0.09
IrrCoef_LateSoy	11	7	0.059
IrrCoef_LateSoy	11	17	0.034
IrrCoef_LateSoy	11	27	0.019
IrrCoef_LateSoy	11	28	0
IrrCoef_LateSoy	12	31	0

Name	Month	Day	Value
IrrCoef_Veg	3	31	0
IrrCoef_Veg	4	1	0.178571
IrrCoef_Veg	8	15	0.178571
IrrCoef_Veg	8	16	0.142857
IrrCoef_Veg	10	31	0.142857
IrrCoef_Veg	11	1	0
IrrCoef_Veg	12	31	0

main.ocl

```
:INCLUDE: OCL\constants.ocl
:INCLUDE: OCL\Forecast-Trigger_Parms.ocl

:Static: statdata.mdb
:Time:   [HomeDir]\basedata\basedata.dss

:if: {[UseForecast]=1}
  :if: {[ForecastData]=cond}
    :Time:   [HomeDir]\basedata\forecasts_cond.dss
  :else:
    :Time:   [HomeDir]\basedata\forecasts_non_cond.dss
  :endif:
:endif:

:MODULE: DLL  AgricDem = modules\AgricDem.DLL

:Include: ocl\udef_list.ocl

:Commands:

:Include: ocl\Agric_calculation.ocl
:Include: ocl\Agric_allocation.ocl
:Include: ocl\inflows.ocl
:Include: ocl\routing.ocl
:Include: ocl\return_flows.ocl
:Include: ocl\drought_plans.ocl

:End:
```

Udef_list.ocl

```
:Udef:

// For agricultural demand calculations and allocations
:FOR: { [county] = { 01, 02, 03, 04, 05, 06, 07, 08, 09, 10 } }

        Udef : dem[county]

:NEXT:

// For inflow filtering
:For:
{   [node] = {010, 040, 070, 100, 150, 190, 220, 250, 410, 415, 420, 440, 550, 600, 650, 700}
}
   Udef : _TempInf[node]
   Udef : _InfDeficit[node]  init{0}

:Next:


/* Set drought plan Udefs */

// Kings Mt Usable Storage
Udef : _KM_Usable_Stor
Udef : _KM_Usable_Stor_Pct init {100}

// For:Next loop to declare drought plan udefs

:FOR: { [Util] = { BRWA, FC, CCW, Shel, KM } }

:FOR: { [level_num] = { 1, 2, 3, 4, 5 } }

   Udef :  _[Util]_Consvn_[level_num]_Demand

   Udef :  _[Util]_Trigger_[level_num]_On         init {0}
        Udef :  _[Util]_Stage_[level_num]_counter         init {0}
        Udef :  _[Util]_Ph_[level_num]_event_counter    init {0}

:NEXT:

:NEXT:

// For use in output tables

:substitute: [level_num] = "1, 2, 3, 4, 5"  // for drought trigger and level determination
```

:substitute: [InflowNd] = "010, 040, 070, 100, 150, 190, 220, 250, 410, 415, 420, 440, 550, 600, 650, 700"

Udef : _CumBoilSpringNatInf init {0}
Udef : _CumGaffneyNatInf init {0}
Udef : _CumBasinNatInf init {0}

// For output files

// Demand nodes
:substitute: [DemandNd] = "012, 042, 052, 061, 072, 086, 102, 152, 186, 192, 222, 224, 252, 412, 416, 436, 442, 442, 552, 554, 602, 604, 645, 652"

// WW returns
:substitute: [WWRet] = "086.150, 086.090, 186.190, 224.250, 306.500, 436.440, 604.610, 606.650, 645.650"

Agric_Calculation.ocl

```
/*  Note the precip data is contained in the basedata file and is based on the record for Tar River Reservoir
(using the Nashville station).
This should be fairly representative of the basin as it is in the middle.  */

// Read in the counties, which are labeled in the Edit Agricultural Data dialog box in the GUI.
:For:
{   [cty] = {01, 02, 03, 04, 05, 06, 07, 08, 09, 10}
}
   RUN_MODULE: AgricDem
   {
    Input:  { [cty],                  // County number
              timesers(Broad/precip),      // Precip in inches

              pattern(IrrCoef_Tobacco),    // Water Use Coefficients for Tobacco, etc.
              pattern(IrrCoef_Turf),
              pattern(IrrCoef_Golf),
              pattern(IrrCoef_ContNurs),
              pattern(IrrCoef_FieldNurs),
              pattern(IrrCoef_Cotton),
              pattern(IrrCoef_EarlySoy),
              pattern(IrrCoef_LateSoy),
              pattern(IrrCoef_Corn),
              pattern(IrrCoef_Veg),
              pattern(IrrCoef_PastHay),
              pattern(IrrCoef_Peanut),
              pattern(IrrCoef_Blueberry),
              pattern(IrrCoef_Strawberry),
              pattern(IrrCoef_Fruit),
              pattern(IrrCoef_Beef),
              pattern(IrrCoef_Dairy),
              pattern(IrrCoef_Horse),
              pattern(IrrCoef_Pig),
              pattern(IrrCoef_Chicken),
              pattern(IrrCoef_Turkey),
              pattern(IrrCoef_OtherAnimal)
             }

    Output: { dem[cty] }
   }
:Next:


/*  The results are in mgd.  Now convert these to acre feet for use in the
agric_allocation.ocl file  */
```

```
Set : dem01 { value : convert_units {dem01, mgd, af } }
Set : dem02 { value : convert_units {dem02, mgd, af } }
Set : dem03 { value : convert_units {dem03, mgd, af } }
Set : dem04 { value : convert_units {dem04, mgd, af } }
Set : dem05 { value : convert_units {dem05, mgd, af } }
Set : dem06 { value : convert_units {dem06, mgd, af } }
Set : dem07 { value : convert_units {dem07, mgd, af } }
Set : dem08 { value : convert_units {dem08, mgd, af } }
Set : dem09 { value : convert_units {dem09, mgd, af } }
Set : dem10 { value : convert_units {dem10, mgd, af } }
```

Agric_allocation.ocl

/* This file allocates the agricultural water demands by the assumed distribution of demand within each reach of interest */

/* The county demand is represented by "dem__" that varies by number. These numbers are established in the agricultural input dialog box, with 01 set forBuncombe, 02 for Cleveland, and so on. (Also see agric_calculation.ocl file and agricultural dialog box) */

Set Demand_Summ_Ag : demand012 { value : dem04 * 0.406 }
Set Demand_Adgr_Ag : demand042 { value : dem04 * 0.355 + dem07 * 0.337 }
Set Demand_Lure_Ag : demand072 { value : dem04 * 0.239 + dem08 * 0.054 + dem01 * 0.921 }
Set Demand_BrGr_Ag : demand102 { value : dem01 * 0.079 + dem08 *0.268 + dem06 * 0.504 + dem07 * 0.627 }

Set Demand_Logn_Ag : demand152 { value : dem06 * 0.35 + dem08 * 0.102 }

Set Demand_2Clf_Ag : demand192 { value : dem08 * 0.235 + dem06 * 0.131 }

Set Demand_Clif_Ag : demand222 { value : dem08 * 0.148 + dem07 * 0.036 + dem10 * 0.0005 }

Set Demand_Boil_Ag : demand252 { value : dem08 * 0.049 }
Set Demand_Casr_Ag : demand412 { value : dem08 * 0.095 }

Set Demand_Lawn_Ag : demand422 { value : dem02 * 0.22 + dem05 * 0.313 + dem08 * 0.05 }

Set Demand_GShl_Ag : demand552 { value : dem02 * 0.487 + dem09 * 0.346 }

Set Demand_KMtn_Ag : demand602 { value : dem05 * 0.688 }

Set Demand_Buff_Ag : demand652 { value : dem02 * 0.292 + dem09 * 0.281 }

inflows.ocl

```
/* Sets the inflows for nodes that need to be filtered. The finalized inflows
   (through September 2009) were already filtered.  However, the provisional inflows
   from the update record routine can be negative due to time of travel or
   imperfect impairment estimations, and therefore are filtered here to prevent
   model infeasibility or unrealistic reservoir releases and/or demand shortages.
*/

:For:
{   [node] = {010, 040, 070, 100, 150, 190, 220, 250, 410, 415, 420, 440, 550, 600, 650, 700}
}
    Set : _TempInf[node] { Value : timesers([node]/inflow) }
    Set : inflow[node] { Value : max{0, _TempInf[node] - _InfDeficit[node](-1) } }
    Set : _InfDeficit[node] { Value : max{0, _InfDeficit[node](-1) - _TempInf[node] } }

:Next:
```

return_flows.ocl

```
/*  Spindale return flows */
Constraint : {dFlow086.150 = lookup {SpindaleWW, month} * dDelivery086}

/*  Ruthefordton return flows */
Constraint : {dFlow086.090 = lookup {RuthefordWW, month} * dDelivery086}

/*  Forest City return flows */
Constraint : {dFlow186.190 = lookup {ForestCityWW, month} * dDelivery186}

/*  Cliffside return flows */
Constraint : {dFlow224.250 = lookup {CliffsideWW, month} * dDelivery224}

/*  Boiling Springs return flows */
Constraint : {dFlow306.500 = lookup {BSpringsWW, month} * dDelivery436}

/* Shelby return flows */
Constraint : {dFlow436.440 = lookup {ShelbyWW, month} * dDelivery436}

/* Kings Mt. return flows */
Constraint : {dFlow604.610 = lookup {KingsMtWW, month} * dDelivery604}

/* Grover return flows */
Constraint : {dFlow606.650 = lookup {GroverWW, month} * dDelivery604}

/* CNA Holdings return flows */
Constraint : {dFlow645.650 = lookup {CNAWW, month} * dDelivery645}routing.ocl
```

routing.ocl

```
/*  File is ROUTING.OCL, which has the coding to handle the Lure -> Boiling Spring,
    and Boiling Springs / Lawndale -> Gaffney, daily time step only. */

/*  1-day lag for each reach.

    Nodes 095 and 525 are reservoirs used for channel storage.
    The flow into these three nodes is unrouted; the release from them is routed.
    The storage in the nodes makes up the difference.

    Use high weights to ensure routing is computed before withdrawals, etc.

    Note that for the first day of the simulation the routed flows need to be
    estimated; the current values set the routed flows equal sum of the natural inflows
    for that day when the simulation is started on 01/01/1929; they may need to be
    adjusted if the run is started on a different date. */


/* Flows from Lure have a lag of 1 day down to Boiling Springs */
Target LureRout : dflow095.100
{   Condition : abs_period <= 1
    priority  : 1
    penalty+  : 100000
    penalty-  : 100000
    value     : convert_units { 67.28, cfs, af }

    Condition : default
    priority  : 1
    penalty+  : 100000
    penalty-  : 100000
    value     : flow090.095(-1)
}

/* Flows from Boiling Springs & the 1st Broad have a lag of
   0.6 * today + 0.4 * yesterday down to Gaffney */
TargetBS_Lawn_Rout : dflow525.550 - 0.6 * dflow500.525
{   Condition : abs_period <= 1
    priority  : 1
    penalty+  : 100000
    penalty-  : 100000
    value     : convert_units { 487, cfs, af }

    Condition : default
    priority  : 1
    penalty+  : 100000
    penalty-  : 100000
```

```
    value     : 0.4 * flow500.525(-1)
}
```

drought_plans.ocl

```
// File is drought_plans.ocl.  Computes trigger levels and demand reductions.

:If: {[Drought_Plans_On] = 1} // First check if drought plan variable is on

/* BRWA */

Set : _BRWA_Trigger_5_On { Value : 0 }

Set : _BRWA_Trigger_4_On
{   Condition : _BRWA_Trigger_4_On(-1) = 1
    {   Condition : flow070.080 + inflow073 > convert_units { 15, mgd, af }
        Value     : 0

        Condition : default
        Value     : _BRWA_Trigger_4_On(-1)
    }

    Condition : _BRWA_Stage_3_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 4
    {   Condition : weekday{year, month, day} <= 1
        {     Condition : flow070.080 + inflow073 < convert_units { 15, mgd, af } and flow070.080(-1) +
inflow073(-1) < convert_units{ 15, mgd, af } and
                    flow070.080(-2) + inflow073(-2) < convert_units { 15, mgd, af } and flow070.080(-3) +
inflow073(-3) < convert_units { 15, mgd, af } and
                    flow070.080(-4) + inflow073(-4) < convert_units { 15, mgd, af } and flow070.080(-5) +
inflow073(-5) < convert_units { 15, mgd, af } and
                    flow070.080(-6) + inflow073(-6) < convert_units { 15, mgd, af }
            Value     : 1
        }
        Condition : default
        Value     : _BRWA_Trigger_4_On(-1)
    }
    Condition : default
    Value     : _BRWA_Trigger_4_On(-1)
}

Set : _BRWA_Trigger_3_On
{   Condition : _BRWA_Trigger_3_On(-1) = 1
    {   Condition : flow070.080 + inflow073 > convert_units { 18, mgd, af }
        Value     : 0

        Condition : default
        Value     : _BRWA_Trigger_3_On(-1)
    }
```

```
   Condition : _BRWA_Stage_2_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 3
   {   Condition : weekday{year, month, day} <= 1
      {     Condition : flow070.080 + inflow073 < convert_units { 18, mgd, af } and flow070.080(-1) +
inflow073(-1) < convert_units { 18, mgd, af } and
                  flow070.080(-2) + inflow073(-2) < convert_units { 18, mgd, af } and flow070.080(-3) +
inflow073(-3) < convert_units { 18, mgd, af } and
                  flow070.080(-4) + inflow073(-4) < convert_units { 18, mgd, af } and flow070.080(-5) +
inflow073(-5) < convert_units { 18, mgd, af } and
                  flow070.080(-6) + inflow073(-6) < convert_units { 18, mgd, af }
         Value     : 1
      }
      Condition : default
      Value     : _BRWA_Trigger_3_On(-1)
   }
   Condition : default
   Value     : _BRWA_Trigger_3_On(-1)
}


Set : _BRWA_Trigger_2_On
{   Condition : _BRWA_Trigger_2_On(-1) = 1
   {   Condition : flow070.080 + inflow073 > convert_units { 20, mgd, af }
      Value     : 0

      Condition : default
      Value     : _BRWA_Trigger_2_On(-1)
   }

   Condition : _BRWA_Stage_1_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 2
   {   Condition : weekday{year, month, day} <= 1
      {     Condition : flow070.080 + inflow073 < convert_units { 20, mgd, af } and flow070.080(-1) +
inflow073(-1) < convert_units { 20, mgd, af } and
                  flow070.080(-2) + inflow073(-2) < convert_units { 20, mgd, af } and flow070.080(-3) +
inflow073(-3) < convert_units { 20, mgd, af } and
                  flow070.080(-4) + inflow073(-4) < convert_units { 20, mgd, af } and flow070.080(-5) +
inflow073(-5) < convert_units { 20, mgd, af } and
                  flow070.080(-6) + inflow073(-6) < convert_units { 20, mgd, af }
         Value     : 1
      }
      Condition : default
      Value     : _BRWA_Trigger_2_On(-1)
   }
   Condition : default
   Value     : _BRWA_Trigger_2_On(-1)
}
```

```
Set : _BRWA_Trigger_1_On
{   Condition : _BRWA_Trigger_1_On(-1) = 1
    {   Condition : flow070.080 + inflow073 > convert_units { 65, mgd, af }
        Value     : 0

        Condition : default
        Value     : _BRWA_Trigger_1_On(-1)
    }


    Condition : weekday{year, month, day} <= 1
    {   Condition : flow070.080 + inflow073 < convert_units { 65, mgd, af } and flow070.080(-1) +
inflow073(-1) < convert_units { 65, mgd, af } and
                flow070.080(-2) + inflow073(-2) < convert_units { 65, mgd, af } and flow070.080(-3) +
inflow073(-3) < convert_units { 65, mgd, af } and
                flow070.080(-4) + inflow073(-4) < convert_units { 65, mgd, af } and flow070.080(-5) +
inflow073(-5) < convert_units { 65, mgd, af } and
                flow070.080(-6) + inflow073(-6) < convert_units { 65, mgd, af }
        Value     : 1

        Condition : default
        Value     : _BRWA_Trigger_1_On(-1)
    }
    Condition : default
    Value     : _BRWA_Trigger_1_On(-1)
}


/* FOREST CITY */

Set : _FC_Trigger_5_On
{   Condition : _FC_Trigger_5_On(-1) = 1
    {   Condition : flow086.150 + inflow150 > convert_units { 5.4, cfs, af }
        Value     : 0

        Condition : default
        Value     : _FC_Trigger_5_On(-1)
    }

    Condition : _FC_Stage_4_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 5
    {   Condition : weekday{year, month, day} <= 1
        {     Condition : flow086.150 + inflow150 < convert_units { 5.4, cfs, af } and flow086.150(-1) +
inflow150(-1) < convert_units { 5.4, cfs, af } and
```

```
                    flow086.150(-2) + inflow150(-2) < convert_units { 5.4, cfs, af } and flow086.150(-3) +
inflow150(-3) < convert_units { 5.4, cfs, af } and
                    flow086.150(-4) + inflow150(-4) < convert_units { 5.4, cfs, af } and flow086.150(-5) +
inflow150(-5) < convert_units { 5.4, cfs, af } and
                    flow086.150(-6) + inflow150(-6) < convert_units { 5.4, cfs, af }
          Value     : 1
      }
      Condition : default
      Value     : _FC_Trigger_5_On(-1)
   }
   Condition : default
   Value     : _FC_Trigger_5_On(-1)
}

Set : _FC_Trigger_4_On
{   Condition : _FC_Trigger_4_On(-1) = 1
   {   Condition : flow086.150 + inflow150 > convert_units { 10.7, cfs, af }
      Value     : 0

      Condition : default
      Value     : _FC_Trigger_4_On(-1)
   }

   Condition : _FC_Stage_3_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 4
   {   Condition : weekday{year, month, day} <= 1
      {     Condition : flow086.150 + inflow150 < convert_units { 10.7, cfs, af } and flow086.150(-1) +
inflow150(-1) < convert_units { 10.7, cfs, af } and
                    flow086.150(-2) + inflow150(-2) < convert_units { 10.7, cfs, af } and flow086.150(-3) +
inflow150(-3) < convert_units { 10.7, cfs, af } and
                    flow086.150(-4) + inflow150(-4) < convert_units { 10.7, cfs, af } and flow086.150(-5) +
inflow150(-5) < convert_units { 10.7, cfs, af } and
                    flow086.150(-6) + inflow150(-6) < convert_units { 10.7, cfs, af }
          Value     : 1
      }
      Condition : default
      Value     : _FC_Trigger_4_On(-1)
   }
   Condition : default
   Value     : _FC_Trigger_4_On(-1)
}

Set : _FC_Trigger_3_On
{   Condition : _FC_Trigger_3_On(-1) = 1
   {   Condition : flow086.150 + inflow150 > convert_units { 16.1, cfs, af }
      Value     : 0
```

```
      Condition : default
      Value     : _FC_Trigger_3_On(-1)
   }

   Condition : _FC_Stage_2_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 3
   {   Condition : weekday{year, month, day} <= 1
      {     Condition : flow086.150 + inflow150 < convert_units { 16.1, cfs, af } and flow086.150(-1) +
inflow150(-1) < convert_units { 16.1, cfs, af } and
                  flow086.150(-2) + inflow150(-2) < convert_units { 16.1, cfs, af } and flow086.150(-3) +
inflow150(-3) < convert_units { 16.1, cfs, af } and
                  flow086.150(-4) + inflow150(-4) < convert_units { 16.1, cfs, af } and flow086.150(-5) +
inflow150(-5) < convert_units { 16.1, cfs, af } and
                  flow086.150(-6) + inflow150(-6) < convert_units { 16.1, cfs, af }
         Value     : 1
      }
      Condition : default
      Value     : _FC_Trigger_3_On(-1)
   }
   Condition : default
   Value     : _FC_Trigger_3_On(-1)
}


Set : _FC_Trigger_2_On
{   Condition : _FC_Trigger_2_On(-1) = 1
   {   Condition : flow086.150 + inflow150 > convert_units { 21.4, cfs, af }
      Value     : 0

      Condition : default
      Value     : _FC_Trigger_2_On(-1)
   }

   Condition : _FC_Stage_1_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 2
   {   Condition : weekday{year, month, day} <= 1
      {     Condition : flow086.150 + inflow150 < convert_units { 21.4, cfs, af } and flow086.150(-1) +
inflow150(-1) < convert_units { 21.4, cfs, af } and
                  flow086.150(-2) + inflow150(-2) < convert_units { 21.4, cfs, af } and flow086.150(-3) +
inflow150(-3) < convert_units { 21.4, cfs, af } and
                  flow086.150(-4) + inflow150(-4) < convert_units { 21.4, cfs, af } and flow086.150(-5) +
inflow150(-5) < convert_units { 21.4, cfs, af } and
                  flow086.150(-6) + inflow150(-6) < convert_units { 21.4, cfs, af }
         Value     : 1
      }
      Condition : default
      Value     : _FC_Trigger_2_On(-1)
```

```
   }
   Condition : default
   Value     : _FC_Trigger_2_On(-1)
}


Set : _FC_Trigger_1_On
{   Condition : _FC_Trigger_1_On(-1) = 1
   {   Condition : flow086.150 + inflow150 > convert_units { 26.8, cfs, af }
      Value     : 0

      Condition : default
      Value     : _FC_Trigger_1_On(-1)
   }


   Condition : weekday{year, month, day} <= 1
   {   Condition : flow086.150 + inflow150 < convert_units { 26.8, cfs, af } and flow086.150(-1) +
inflow150(-1) < convert_units { 26.8, cfs, af } and
            flow086.150(-2) + inflow150(-2) < convert_units { 26.8, cfs, af } and flow086.150(-3) +
inflow150(-3) < convert_units { 26.8, cfs, af } and
            flow086.150(-4) + inflow150(-4) < convert_units { 26.8, cfs, af } and flow086.150(-5) +
inflow150(-5) < convert_units { 26.8, cfs, af } and
            flow086.150(-6) + inflow150(-6) < convert_units { 26.8, cfs, af }
      Value     : 1

      Condition : default
      Value     : _FC_Trigger_1_On(-1)
   }
   Condition : default
   Value     : _FC_Trigger_1_On(-1)
}


/* CLEVELAND COUNTY WATER */

Set : _CCW_Trigger_5_On
{   Condition : _CCW_Trigger_5_On(-1) = 1
   {   Condition : inflow410 > convert_units { 5, cfs, af }
      Value     : 0

      Condition : default
      Value     : _CCW_Trigger_5_On(-1)
   }

   Condition : _CCW_Stage_4_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 5
```

```
   {   Condition : weekday{year, month, day} < 5
      {     Condition : inflow410 < convert_units { 5, cfs, af }
            Value     : 1

        Condition : default
        Value     : _CCW_Trigger_5_On(-1)
        }
   }
   Condition : default
   Value     : _CCW_Trigger_5_On(-1)
}

Set : _CCW_Trigger_4_On
{   Condition : _CCW_Trigger_4_On(-1) = 1
   {   Condition : inflow410 > convert_units { 5, cfs, af }
       Value     : 0

       Condition : default
       Value     : _CCW_Trigger_4_On(-1)
   }

   Condition : _CCW_Stage_3_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 4
   {   Condition : weekday{year, month, day} <= 1
       {     Condition : inflow410 <= convert_units { 5, cfs, af } and inflow410(-1) <= convert_units { 5,
cfs, af } and
                     inflow410(-2) <= convert_units { 5, cfs, af } and inflow410(-3) <= convert_units { 5, cfs,
af } and
                     inflow410(-4) <= convert_units { 5, cfs, af }
            Value     : 1
       }
       Condition : default
       Value     : _CCW_Trigger_4_On(-1)
   }
   Condition : default
   Value     : _CCW_Trigger_4_On(-1)
}

Set : _CCW_Trigger_3_On
{   Condition : _CCW_Trigger_3_On(-1) = 1
   {   Condition : inflow410 > convert_units { 10, cfs, af }
       Value     : 0

       Condition : default
       Value     : _CCW_Trigger_3_On(-1)
   }
```

```
   Condition : _CCW_Stage_2_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 3
   {   Condition : weekday{year, month, day} <= 1
      {     Condition : inflow410 < convert_units { 10, cfs, af } and inflow410(-1) < convert_units { 10,
cfs, af } and
                  inflow410(-2) < convert_units { 10, cfs, af } and inflow410(-3) < convert_units { 10, cfs,
af } and
                  inflow410(-4) < convert_units { 10, cfs, af } and inflow410(-5) < convert_units { 10, cfs,
af } and
                  inflow410(-6) < convert_units { 10, cfs, af }
          Value     : 1
      }
      Condition : default
      Value     : _CCW_Trigger_3_On(-1)
   }
   Condition : default
   Value     : _CCW_Trigger_3_On(-1)
}


Set : _CCW_Trigger_2_On
{   Condition : _CCW_Trigger_2_On(-1) = 1
   {   Condition : inflow410 > convert_units { 15, cfs, af }
      Value     : 0

      Condition : default
      Value     : _CCW_Trigger_2_On(-1)
   }

   Condition : _CCW_Stage_1_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 2
   {   Condition : weekday{year, month, day} <= 1
      {     Condition : inflow410 < convert_units { 15, cfs, af } and inflow410(-1) < convert_units { 15,
cfs, af } and
                  inflow410(-2) < convert_units { 15, cfs, af } and inflow410(-3) < convert_units { 15, cfs,
af } and
                  inflow410(-4) < convert_units { 15, cfs, af } and inflow410(-5) < convert_units { 15, cfs,
af } and
                  inflow410(-6) < convert_units { 15, cfs, af } and inflow410(-7) < convert_units { 15, cfs,
af } and
                  inflow410(-8) < convert_units { 15, cfs, af } and inflow410(-9) < convert_units { 15, cfs,
af }
         Value     : 1
      }
      Condition : default
      Value     : _CCW_Trigger_2_On(-1)
   }
```

```
   Condition : default
   Value     : _CCW_Trigger_2_On(-1)
}


Set : _CCW_Trigger_1_On
{   Condition : _CCW_Trigger_1_On(-1) = 1
   {   Condition : inflow410 > convert_units { 25, cfs, af }
       Value     : 0

       Condition : default
       Value     : _CCW_Trigger_1_On(-1)
   }


   Condition : weekday{year, month, day} <= 1
   {   Condition : inflow410 < convert_units { 25, cfs, af } and inflow410(-1) < convert_units { 25, cfs, af
} and
                 inflow410(-2) < convert_units { 25, cfs, af } and inflow410(-3) < convert_units { 25, cfs,
af } and
                 inflow410(-4) < convert_units { 25, cfs, af } and inflow410(-5) < convert_units { 25, cfs,
af } and
                 inflow410(-6) < convert_units { 25, cfs, af } and inflow410(-7) < convert_units { 25, cfs,
af } and
                 inflow410(-8) < convert_units { 25, cfs, af } and inflow410(-9) < convert_units { 25, cfs,
af } and
                 inflow410(-10) < convert_units { 25, cfs, af } and inflow410(-11) < convert_units { 25,
cfs, af } and
                 inflow410(-12) < convert_units { 25, cfs, af } and inflow410(-13) < convert_units { 25,
cfs, af } and
                 inflow410(-14) < convert_units { 25, cfs, af }
       Value     : 1

       Condition : default
       Value     : _CCW_Trigger_1_On(-1)
   }
   Condition : default
   Value     : _CCW_Trigger_1_On(-1)
}


/* SHELBY */

Set : _Shel_Trigger_5_On
{   Condition : _Shel_Trigger_5_On(-1) = 1
   {   Condition : flow415.420 + inflow420 > convert_units { 25, cfs, af }
       Value     : 0
```

```
      Condition : default
      Value     : _Shel_Trigger_5_On(-1)
   }

   Condition : _Shel_Stage_4_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 3
   {   Condition : weekday{year, month, day} <= 1
      {     Condition : flow260.430(-1) >= convert_units { 9, mgd, af }
            Value     : 1
      }
      Condition : default
      Value     : _Shel_Trigger_5_On(-1)
   }
   Condition : default
   Value     : _Shel_Trigger_5_On(-1)
}

Set : _Shel_Trigger_4_On
{   Condition : _Shel_Trigger_4_On(-1) = 1
   {   Condition : flow415.420 + inflow420 > convert_units { 25, cfs, af }
      Value     : 0

      Condition : default
      Value     : _Shel_Trigger_4_On(-1)
   }

   Condition : _Shel_Stage_3_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 3
   {   Condition : weekday{year, month, day} <= 1
      {     Condition : flow415.420 + inflow420 < convert_units { 25, cfs, af } and flow260.430(-1) >
convert_units { 7.2, mgd, af }
            Value     : 1
      }
      Condition : default
      Value     : _Shel_Trigger_4_On(-1)
   }
   Condition : default
   Value     : _Shel_Trigger_4_On(-1)
}

Set : _Shel_Trigger_3_On
{   Condition : _Shel_Trigger_3_On(-1) = 1
   {   Condition : flow415.420 + inflow420 > convert_units { 25, cfs, af }
      Value     : 0

      Condition : default
```

```
      Value     : _Shel_Trigger_3_On(-1)
   }

   Condition : _Shel_Stage_2_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 3
   {   Condition : weekday{year, month, day} <= 1
      {     Condition : flow415.420 + inflow420 < convert_units { 25, cfs, af } and flow415.420(-1) +
inflow420(-1) < convert_units { 25, cfs, af } and
                  flow415.420(-2) + inflow420(-2) < convert_units { 25, cfs, af } and flow415.420(-3) +
inflow420(-3) < convert_units { 25, cfs, af } and
                  flow415.420(-4) + inflow420(-4) < convert_units { 25, cfs, af } and flow415.420(-5) +
inflow420(-5) < convert_units { 25, cfs, af } and
                  flow415.420(-6) + inflow420(-6) < convert_units { 25, cfs, af } and flow415.420(-7) +
inflow420(-7) < convert_units { 25, cfs, af } and
                  flow415.420(-8) + inflow420(-8) < convert_units { 25, cfs, af } and flow415.420(-9) +
inflow420(-9) < convert_units { 25, cfs, af } and
                  flow415.420(-10) + inflow420(-10) < convert_units { 25, cfs, af } and flow415.420(-11)
+ inflow420(-11) < convert_units { 25, cfs, af } and
                  flow415.420(-12) + inflow420(-12) < convert_units { 25, cfs, af } and flow415.420(-13)
+ inflow420(-13) < convert_units { 25, cfs, af } and
                  flow415.420(-14) + inflow420(-14) < convert_units { 25, cfs, af }
         Value     : 1
      }
      Condition : default
      Value     : _Shel_Trigger_3_On(-1)
   }
   Condition : default
   Value     : _Shel_Trigger_3_On(-1)
}


Set : _Shel_Trigger_2_On
{   Condition : _Shel_Trigger_2_On(-1) = 1
   {   Condition : flow420.440 > convert_units { 25, cfs, af }
      Value     : 0

      Condition : default
      Value     : _Shel_Trigger_2_On(-1)
   }

   Condition : _Shel_Stage_1_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 2
   {   Condition : weekday{year, month, day} <= 1
      {   Condition : flow420.440 <= convert_units { 25, cfs, af } and flow420.440(-1) < convert_units {
21.4, cfs, af } and
                  flow420.440(-2) <= convert_units { 25, cfs, af } and flow420.440(-3) <= convert_units {
25, cfs, af } and
```

```
                flow420.440(-4) <= convert_units { 25, cfs, af } and flow420.440(-5) <= convert_units {
25, cfs, af } and
                flow420.440(-6) <= convert_units { 25, cfs, af }
          Value     : 1
       }
       Condition : default
       Value     : _Shel_Trigger_2_On(-1)
   }
   Condition : default
   Value     : _Shel_Trigger_2_On(-1)
}


Set : _Shel_Trigger_1_On
{   Condition : _Shel_Trigger_1_On(-1) = 1
   {   Condition : flow415.420 + inflow420 > convert_units { 30, cfs, af }
       Value     : 0

       Condition : default
       Value     : _Shel_Trigger_1_On(-1)
   }


   Condition : weekday{year, month, day} <= 1
   {   Condition : flow415.420 + inflow420 < convert_units { 30, cfs, af } and flow415.420(-1) +
inflow420(-1) < convert_units { 30, cfs, af } and
                flow415.420(-2) + inflow420(-2) < convert_units { 30, cfs, af } and flow415.420(-3) +
inflow420(-3) < convert_units { 30, cfs, af } and
                flow415.420(-4) + inflow420(-4) < convert_units { 30, cfs, af } and flow415.420(-5) +
inflow420(-5) < convert_units { 30, cfs, af } and
                flow415.420(-6) + inflow420(-6) < convert_units { 30, cfs, af }
       Value     : 1

       Condition : default
       Value     : _Shel_Trigger_1_On(-1)
   }
   Condition : default
   Value     : _Shel_Trigger_1_On(-1)
}

/* KINGS MT */

Set : _KM_Usable_Stor { Value : max { 0, (storage600 - lower_rule600) } }
Set : _KM_Usable_Stor_Pct { Value :  _KM_Usable_Stor / (upper_rule600 - lower_rule600)  * 100 }

Set : _KM_Trigger_5_On
{   Condition : _KM_Trigger_5_On(-1) = 1
```

```
   {   Condition : _KM_Usable_Stor_Pct >= 100
       Value     : 0

       Condition : default
       Value     : _KM_Trigger_5_On(-1)
   }

   Condition : _KM_Stage_4_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 5
   {   Condition : weekday{year, month, day} <= 1
       {     Condition : _KM_Usable_Stor_Pct < 10
             Value     : 1
       }
       Condition : default
       Value     : _KM_Trigger_5_On(-1)
   }
   Condition : default
   Value     : _KM_Trigger_5_On(-1)
}

Set : _KM_Trigger_4_On
{   Condition : _KM_Trigger_4_On(-1) = 1
   {   Condition : _KM_Usable_Stor_Pct >= 100
       Value     : 0

       Condition : default
       Value     : _KM_Trigger_4_On(-1)
   }

   Condition : _KM_Stage_3_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 4
   {   Condition : weekday{year, month, day} <= 1
       {     Condition : _KM_Usable_Stor_Pct < 30
             Value     : 1
       }
       Condition : default
       Value     : _KM_Trigger_4_On(-1)
   }
   Condition : default
   Value     : _KM_Trigger_4_On(-1)
}

Set : _KM_Trigger_3_On
{   Condition : _KM_Trigger_3_On(-1) = 1
   {   Condition : _KM_Usable_Stor_Pct >= 100
       Value     : 0
```

```
      Condition : default
      Value     : _KM_Trigger_3_On(-1)
   }

   Condition : _KM_Stage_2_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 3
   {   Condition : weekday{year, month, day} <= 1
      {     Condition : _KM_Usable_Stor_Pct < 50
           Value     : 1
      }
      Condition : default
      Value     : _KM_Trigger_3_On(-1)
   }
   Condition : default
   Value     : _KM_Trigger_3_On(-1)
}


Set : _KM_Trigger_2_On
{   Condition : _KM_Trigger_2_On(-1) = 1
   {   Condition : _KM_Usable_Stor_Pct >= 100
      Value     : 0

      Condition : default
      Value     : _KM_Trigger_2_On(-1)
   }

   Condition : _KM_Stage_1_Counter(-1) >= 1   // Require a 1 day (can be changed if utilities want)
waiting period before going into phase 2
   {   Condition : weekday{year, month, day} <= 1
      {     Condition : _KM_Usable_Stor_Pct < 65
         Value     : 1
      }
      Condition : default
      Value     : _KM_Trigger_2_On(-1)
   }
   Condition : default
   Value     : _KM_Trigger_2_On(-1)
}


Set : _KM_Trigger_1_On
{   Condition : _KM_Trigger_1_On(-1) = 1
   {   Condition : _KM_Usable_Stor_Pct >= 100
      Value     : 0

      Condition : default
```

```
      Value     : _KM_Trigger_1_On(-1)
   }


   Condition : weekday{year, month, day} <= 1
   {   Condition : _KM_Usable_Stor_Pct < 75
       Value     : 1

       Condition : default
       Value     : _KM_Trigger_1_On(-1)
   }
   Condition : default
   Value     : _KM_Trigger_1_On(-1)
}



/*  This section sets/resets the counters used to maintain the proper spacing of conservation stages */

:For:
{ [Util] = { BRWA, FC, CCW, Shel, KM }
}

:For:
{   [trig] = {1, 2, 3, 4, 5}
}

Set : _[Util]_Stage_[trig]_Counter
{   Condition : _[Util]_Stage_[trig]_Counter(-1) > 0 and _[Util]_Trigger_[trig]_On = 0
    Value     : 0

    Condition : _[Util]_Trigger_[trig]_On = 1
    Value     : _[Util]_Stage_[trig]_Counter(-1) + 1

    Condition : default
    Value     : _[Util]_Stage_[trig]_Counter(-1)
}

/*  Count all trigger events lasting at least 7 days */

Set : _[Util]_Ph_[trig]_event_counter
{   Condition : _[Util]_Stage_[trig]_Counter = 7 and _[Util]_Stage_[trig]_Counter(-1) = 6
    Value     : _[Util]_Ph_[trig]_event_counter(-1) + 1

    Condition : default
    Value      : _[Util]_Ph_[trig]_event_counter(-1)
}
```

```
:Next:

:Next:

// Set conservation demand and delivery constraints for each trigger level in effect.

// Set demand reduction factors
:SUBSTITUTE: [Dem_1_Red_Factor] = 5
:SUBSTITUTE: [Dem_2_Red_Factor] = 10
:SUBSTITUTE: [Dem_3_Red_Factor] = 20
:SUBSTITUTE: [Dem_4_Red_Factor] = 25
:SUBSTITUTE: [Dem_5_Red_Factor] = 35

:For:
{   [trig] = {1, 2, 3, 4, 5}
}

// BRWA
Set : _BRWA_Consvn_[trig]_Demand
{   Condition : _BRWA_Trigger_[trig]_On = 1
    Value     : Demand086 *  ( 1 - [Dem_[trig]_Red_Factor] / 100 )

    Condition : default
    Value     : Demand086
}

Constraint   BRWA_Demand_Limit_Consvn_[trig] :
          {   Condition  :  _BRWA_Trigger_[trig]_On = 1
              Expression :  dflow080.086 <= _BRWA_Consvn_[trig]_Demand }

// Forest City
Set : _FC_Consvn_[trig]_Demand
{   Condition : _FC_Trigger_[trig]_On = 1
    Value     : Demand186 *  ( 1 - [Dem_[trig]_Red_Factor] / 100 )

    Condition : default
    Value     : Demand186
}

Constraint   FC_Demand_Limit_Consvn_[trig] :
          {   Condition  :  _FC_Trigger_[trig]_On = 1
              Expression :  dflow150.186 <= _FC_Consvn_[trig]_Demand }

// CCW
Set : _CCW_Consvn_[trig]_Demand
{   Condition : _CCW_Trigger_[trig]_On = 1
```

```
   Value     : Demand416 *  ( 1 - [Dem_[trig]_Red_Factor] / 100 )

   Condition : default
   Value     : Demand416
}

Constraint   CCW_Demand_Limit_Consvn_[trig] :
         {   Condition  :  _CCW_Trigger_[trig]_On = 1
             Expression :  dflow410.416 <= _CCW_Consvn_[trig]_Demand }

// Shelby
Set : _Shel_Consvn_[trig]_Demand
{   Condition : _Shel_Trigger_[trig]_On = 1
   Value     : Demand436 *  ( 1 - [Dem_[trig]_Red_Factor] / 100 )

   Condition : default
   Value     : Demand436
}

Constraint   Shel_Demand_Limit_Consvn_[trig] :
         {   Condition  :  _Shel_Trigger_[trig]_On = 1
             Expression :  dflow430.436 <= _Shel_Consvn_[trig]_Demand }

// Kings Mt
Set : _KM_Consvn_[trig]_Demand
{   Condition : _KM_Trigger_[trig]_On = 1
   Value     : Demand604 *  ( 1 - [Dem_[trig]_Red_Factor] / 100 )

   Condition : default
   Value     : Demand604
}

Constraint   KM_Demand_Limit_Consvn_[trig] :
         {   Condition  :  _KM_Trigger_[trig]_On = 1
             Expression :  dflow601.604 <= _KM_Consvn_[trig]_Demand }

:Next:


:else:


:endif:
```

APPENDIX B –

Finalized Inflow Data Development

Section 1. Introduction

This report provides a detailed account of the inflow development for the Broad River Basin Hydrologic Model. The inflow record runs from January 1930 to September 2009[1]. This period is designed to capture as many drought events as possible, including the extreme droughts in 2002 and 2007. There are 8 streamflow gages in the basin that are used in this project as well as 4 reference gages outside of the basin. These are listed in Table 1. These gages have at least 10 years of daily data with which to make valid statistical comparisons with other gages. Most of the gages have incomplete records; they either started after 1930 or ended before 2009. Some of the gages were used just to provide more data for *fillin* (see below) when computing statistics.

The inflow dataset is based on "unimpaired" gage flows. Gages only show the actual flow in the stream; they have no information about what the flow would have been without human intervention. "Impairments" are modifications of the natural flows due to change in reservoir storage (including evaporation and precipitation on the reservoir surface) and consumptive withdrawals of water (municipal, industrial, or agricultural). If water is withdrawn above a gage and returned to the river below the gage, the impairment is the entire withdrawal.

The next section describes the process used to compute daily flows and gains. Because of the noise in the data, it is important to look at the data at each step to find unrealistic values. These are noted later.

[1] A provisional record extends beyond this date, but this does not account for most of the actual impairments. Future updates will require impairment data for the inflow dataset to be considered finalized.

Section 2. Data and General Procedure

The first step in building the record is to compute the unimpaired gage flows. These computations are contained in the spreadsheet ***inflow_unimpairment.xls*** in the 'Inflow Spreadsheets' direcotry. The unimpaired gage data is summarized in the ***unimpaired_summary.xls*** file. Impairments in the basin accumulate as each downstream gage is included. For example, the impairments upstream of Boiling Springs include the impairments on the Broad River (including the operations of dams upstream), the Second Broad River, and the Green River. The impairment is calculated as follows:

Unimpaired gage flow = gage flow + upstream water withdrawal (by agricultural, municipal, and industrial users) – upstream discharge (water or wastewater from municipal or industrial users, including power plants) + upstream change in reservoir storage + upstream evaporation on the reservoir surface - upstream precipitation on the reservoir surface.

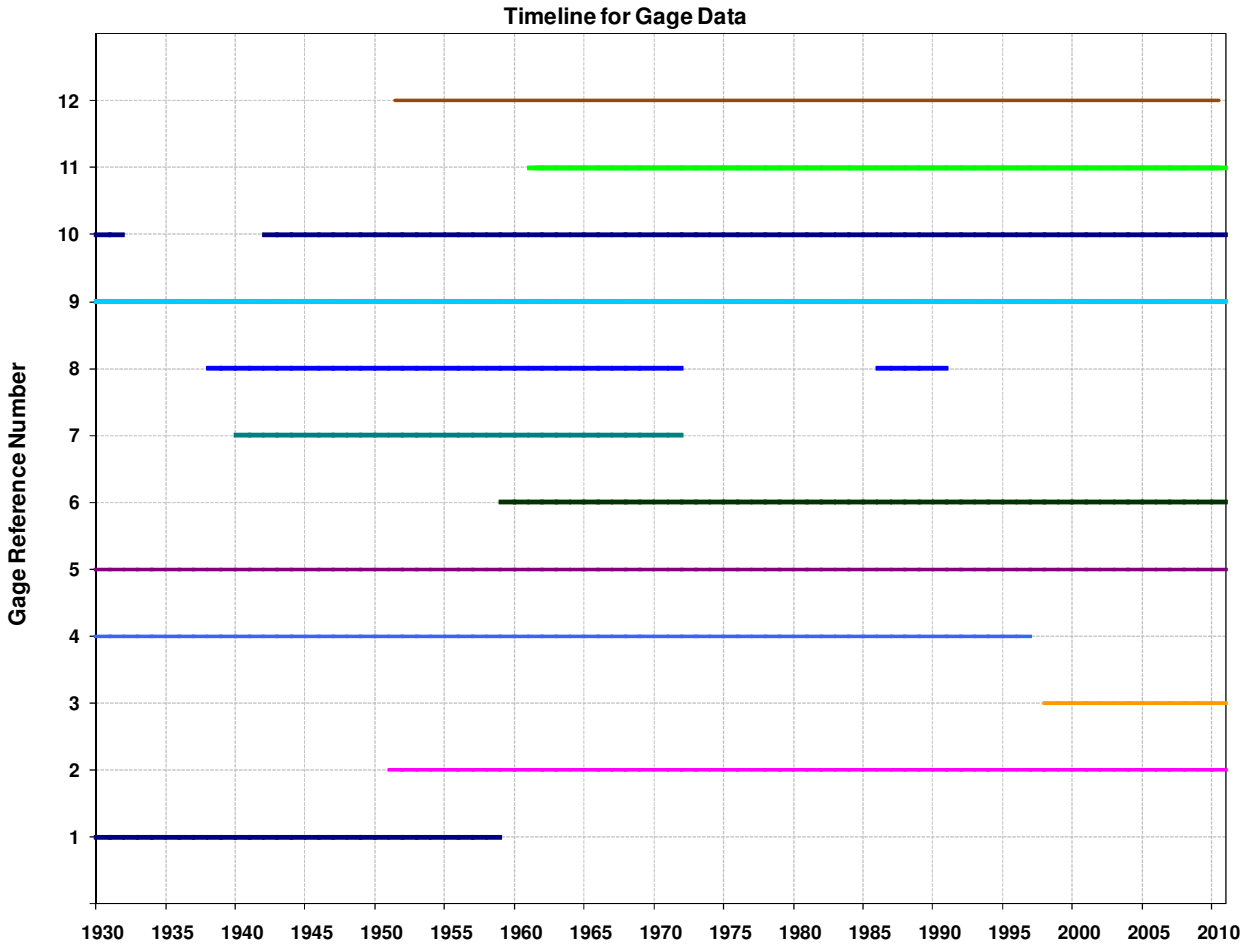
The discharge and withdrawal data were obtained by Moffatt and Nichol. The data are based on monthly average data for recent years, and extrapolated back to 1930 using population census data.

Evaporation and precipitation data were also collected by HydroLogics. Evaporation data are based NOAA annual evaporation maps, which is disaggregated to monthly values using a NOAA pan evaporation distribution study[2], resulting in monthly pattern of evaporation on each reservoir surface. Precipitation data from the station closest to a reservoir are used to estimate the precipitation at the reservoir. When these data are missing, data from the next nearest station are used, and so on. COOP stations used include Tyron, Lake Lure, Caroleen, Simms, Marion, Casar, Shelby, Gaston Shoals, and Gaffney. For each reservoir, HydroLogics calculated a daily timeseries of net evaporation (or the difference between evaporation and precipitation) for the hydrologic record. These data are contained in spreadsheets in the "Evap-Precip" folder. These data are used to (1) estimate the historic change in reservoir storage due to net evaporation and (2) estimate net evaporation on the reservoir surface during OASIS model simulation.

[2] "Broad River CHEOPS Model Operations Report," DTA, December 2007.

Table 1. List of Gages

USGS Number	Description	Period of Record	Ref No.	Ref. Name	Drain. Area
Basin Gages					
02148500	BROAD RIVER NEAR CHIMNEY ROCK, NC	04/1927 - 09/1958	1	Lure	97
02149000	COVE CREEK NEAR LAKE LURE, NC	01/1951 - present	2	Cove	79
02150495	SECOND BROAD RIVER NEAR LOGAN, NC	10/1998 - present	3	Logn	86.2
02151000	SECOND BROAD RIVER AT CLIFFSIDE, NC	07/1925 - 12/1996	4	Clif	220
02151500	BROAD RIVER NEAR BOILING SPRINGS, NC	07/1925 - present	5	Boil	875
02152100	FIRST BROAD RIVER NEAR CASAR, NC	03/1959 - present	6	Casr	60.5
02152500	FIRST BROAD RIVER NEAR LAWNDALE, NC	03/1940 - 09/1971	7	Lawn	200
02153500	BROAD RIVER NEAR GAFFNEY, NC	12/1938 - 09/1971, 06/1986 - 09/1990	8	Gaff	1,490
Reference gages outside of the basin					
02138500	LINVILLE RIVER NEAR NEBO, NC	07/1922 - present	9	Nebo	66.7
02143000	HENRY FORK NEAR HENRY RIVER, NC	08/1925 - present	10	Hnry	83.2
02143040	JACOB FORK AT RAMSEY, NC	10/1961 - present	11	Jacb	25.7
02143500	INDIAN CREEK NEAR LABORATORY, NC	09/1951 - present	12	Indn	69.2



The second step in the inflow development process is to fill in the missing flows and gains for each gage with missing records. This requires assembling a monthly record of unimpaired flows and gains based on the daily unimpaired data computed above. These flows and gains are fed into a program named *fillin* (developed by William Alley and Alan Burns of the USGS[3]). We will refer to these as "extended" flows and gains. This is done on a monthly basis because *fillin* only works with monthly data. The gages associated with the flows and gains used in the remainder of this document are shown in Table 2. The third step is to apportion the extended flows and gains to make sure that their volumes match downstream unimpaired gage flows. The monthly flows and gains are then disaggregated into daily values using local, unimpaired gages. These steps are described in detail in Section 3.

The last step in the process is to compute the OASIS nodal inflows based on the flows and gains computed above. This step is described in detail in Section 4.

[3] "Mixed-Station Extension of Monthly Streamflow Records," *Journal of Hydraulic Engineering*, ASCE, Vol. 109, No. 10, October 1983.

For the Broad River near Chimney Rock gage (Lure), *fillin* was used to complete its missing record from October 1958 to September 2009.

For the Second Broad River near Logan (Logn), *fillin* was used to complete its missing record from January 1930 to September 1998.

For the Second Broad River at Cliffside (Cliff), *fillin* was used to complete its missing record from January 1997 to September 2009.

For the First Broad River near Lawndale gage (Lawn), *fillin* was used to complete its missing record from January 1930 to February 1940, and from October 1971 to September 2009.

For the Broad River near Gaffney gage (Gaff), *fillin* was used to complete its missing record from January 1930 to November 1938, from October 1971 to May 1986, and from October 1990 to January 2009.

Table 2. Gages Where Flows and Gains Are Computed

Gage	Flow or Gain	Gain Calculation
Lure	Flow	
Logn	Flow	
Cliff	Flow	
Boil	Gain	Boil Flow – Cliff Flow – Lure Flow
Lawn	Flow	
Gaff	Gain	Gaff Flow – Lawn Flow – Boil Flow

Section 3. Computation of Extended Gage Flows and Reach Gains

All the computations outlined in this section are done on monthly data, which reduces noise and is required for statistical hydrology programs like *fillin*. Noisy data occurs when time of travel differences occur or when the impairment data create artificial variation in the flows.

First, the actual gains are determined from the unimpaired gage flows. These computations are done with the DSSVue script, ***compute_gain.py.*** Next, *fillin* is run to compute the extended flows and gains for the gages with missing records. Note that *fillin* preserves the actual flows and gains where they exist. These extended flows and gains are then "scaled." The objective of scaling is to ensure that the sum of filled-in flows upstream of a gage with an actual record equals the actual unimpaired flow at that gage. The *fillin* program does not ensure this for two reasons. First, it utilizes only a single correlated record for each value generated, thus ignoring sums, and second, it works with log transforms, and not actual flows.

Here is an example. We want to compute the unimpaired flow at the Cliffside gage (Clif) when the gage data is missing. We extended the gain at Clif using *fillin*. Now we want to adjust those extended values so that the sum of the flows and gains down to the Boiling Springs (Boil) match the unimpaired flow at the gage. So we say that we maintain the Boil flow by scaling with the sum of the Clif and Lure extended flows. The calculation is:

Scaled Cliff extended flow = (Boil flow)* (Clif extended flow) / (Clif extended flow + Lure extended flow + Boil extended gain)

The companion calculation for the flow at Boil is:

Scaled Boil extended gain = (Boil flow)* (Boil extended gain) / (Clif extended flow + Lure extended flow + Boil extended gain)

Thus the sum of the scaled flows/gains (Clif, Lure, and Boil) equals the unimpaired gage flow at Boil.

In this way we ensure that the total volume of all the flows and gains, be they actual or extended, upstream from a given gage match the unimpaired flow at the gage, preserving the unimpaired gaged flows.

The DSSVue script used to do these calculations is ***scale_flow_gain.py***. The output is the file ***scale_flow_gain.dss***, which contains monthly flows and gains.

The next step is to disaggregate the monthly flows into daily flows. This is done using flows for a daily, unimpaired gage that is local or has similar drainage area (call it a "reference gage") along with our monthly flows. We multiply the monthly value by the ratio of that day's flow to that month's flow at the reference gage. The disaggregation formula is:

daily ratio = daily reference value / monthly reference value
daily computed value = monthly computed value * daily ratio

Two DSSVue scripts are used to do this step: ***convert_month_to_day.py*** and ***disaggregate.py***. DSSVue cannot work on two records with different time steps, so the first step is to convert the monthly value to daily. For example, the gain at Boil for January 1930 is 912 cfs; converting the monthly flow to daily gives 912 cfs for each day in the month. The first script converts the monthly flows to daily flows for each location where we need them. The second script computes the daily flows and gains as shown above.

It is important to note that we are not trying to replicate history in computing the OASIS inflows; rather, we are trying to build daily flows whose variation is *representative* of history while preserving unimpaired gaged flows as "ground truth".

Note that actual daily values of unimpaired gage *flows* are often maintained in the script files. Actual daily *gains* between gages are generally not maintained due to their "noise", so the script files aggregate them monthly and then disaggregate them back to daily values using a locally unimpaired gage. Therefore, actual flows and gains on a monthly basis are maintained, but generally the former will only be maintained on a daily basis.

The remainder of this section shows the details in how the records were extended and disaggregated. This is the short hand used in the remainder of this document:

d/s = downstream
u/s = upstream
DA = drainage area
F = actual or scaled flow at a gage
XF = "extended" flow at a gage as computed by *fillin* when actual flows do not exist
G = actual or scaled gain, or inflow, between two locations, which is the difference of u/s gage F and d/s gage F
XG = extended gain between two locations as computed by *fillin* when actual gains do not exist

LAKE LURE (CHIMNEY ROCK) FLOW

10/1928 – 09/1958
Use actual monthly values

10/1958 – 12/1996
LureF = (BoilF – ClifF) * LureXF / (BoilXG + LureXF)

01/1997 – present
LureF = (BoilF) * LureXF / (BoilXG + CliffXF + Lure XF)

Disaggregate to daily using Cove/Clif.

2ND BROAD CLIFFSIDE FLOW

01/1997 – present
ClifF = (BoilF) * ClifXF/ (BoilXG + CliffXF - LureXF)

Disaggregate to daily using Cove.

BOILING SPRINGS GAIN

10/1958 – 12/1996
BoilG = (BoilF – ClifF) * BoilXG / (BoilXG + LureXF)

01/1997 – present
BoilG = (BoilF) * BoilXG / (BoilXG + CliffXF + Lure XF)

Disaggregate to daily using Clif/Cove

LAWNDALE FLOW
10/1928 – 09/1938 – No Gaffney gage to scale to.

10/1938 – 02/1940
LawnF = (GaffF – BoilF) * LawnXF / (GaffXG + LawnXF)

03/1940 – 09/1971 – use actual flow

09/1971 – 05/1986 – No Gaffney to scale to, use filled in gain.

06/1986 – 06/1990
LawnF = (GaffF – BoilF) * LawnXF / (GaffXG + LawnXF)

07/1990 – present – No Gaffney to scale to, use filled in gain.

Disaggregate to daily using Casr/Cliff/Hnry

GAFFNEY GAIN
10/1928 – 11/1938 – No Gaffney gage to scale to.

12/1938 – 02/1940
GaffG = (GaffF – BoilF) * GaffXG / (GaffXG + LawnXF)

03/1940 – 09/1971 – use actual gain

09/1971 – 05/1986 – No Gaffney to scale to, use filled in gain.

06/1986 – 06/1990
GaffG = (GaffF – BoilF) * GaffXG / (GaffXG + LawnXF)

07/1990 – present – No Gaffney to scale to, use filled in gain.

Disaggregate to daily using Indian Ck.

Section 4. Computing Inflows at OASIS Nodes from the Flows and Gains

This section describes the computation of inflows at OASIS nodes from flows and gains at gages described above. The computations are done "on the fly" during an OASIS run called "Compute_Inflows" using ratios of drainage area (total or incremental) as shown in Table 3. The computations are included in an OCL file called *set_inflows.ocl* and are listed below.

Node 010 (Lake Summit)	CoveF * 42.4 / 79
Node 040 (Lake Adger)	CoveF * 94.3 / 79
Node 070 (Lake Lure)	LureF
Node 150 (2nd Broad FC)	LognF * 92 / 86.2
Node 190 (2nd Broad Cliffside)	ClifF – inflow150
Node 100 (Green-Broad Conf)	(BoilG – inflow010 – inflow040) * 339 / 424
Node 220 (Cliffside Plant)	(BoilG – inflow010 – inflow040) * 56 / 424
Node 250 (Boiling Springs)	(BoilG – inflow010 – inflow040) * 29 / 424
Node 410 (1st Broad CCW)	LawnF * 181 / 200
Node 415 (1st Broad Lawndale)	LawnF * 19 / 200
Node 420 (1st Broad Shelby)	LawnF * 74 / 200
Node 440 (Stice Shoals)	LawnF * 33 / 200
Node 600 (Kings Mtn)	IndnF * 67.5 / 69.2
Node 650 (Buffalo Ck)	IndnF * 107.3 / 69.2
Node 550 (Gaston Shoals)	(GaffG – inflow420 – inflow440 – inflow600 – inflow650) * 1297 / 1490
Node 700 (Gaffney)	(GaffG – inflow420 – inflow440 – inflow600 – inflow650) * 193 / 1490

Table 3. List of Drainage Areas (in square miles)

Reservoirs			Gages		
Description	Total Drainage Area	Incremental Drainage Area	Description	Total Drainage Area	Incremental Drainage Area
Lake Lure	94.2	-	BROAD RIVER NEAR CHIMNEY ROCK, NC	97	-
Lake Summit	42.4	-	SECOND BROAD RIVER NEAR LOGAN, NC	86.2	-
Lake Adger	136.7	94.3	SECOND BROAD RIVER AT CLIFFSIDE, NC	220	133.8
Cliffside Dam	846	392.3	BROAD RIVER NEAR BOILING SPRINGS, NC	875	29
Stice Shoals	288	88	FIRST BROAD RIVER NEAR LAWNDALE, NC	200	-
Kings Mt. Dam	67.5	-	BROAD RIVER NEAR GAFFNEY, NC	1,490	125.5
Gaston Shoals	1,297	134			

Section 5. Error Checking and Inflow Filtering

As noted in Section 3, because of the noisy data, a lot of error checking is necessary. These are some of the errors that can occur.

- Negative unimpaired gage flow. These are physically impossible and should be corrected.

- Negative gains. These are sometimes legitimate. However, there are times when a flood hits a gage at the very end of the month, while not arriving at the gage downstream until the beginning of the next month. This can cause a highly negative gain in the first month and a highly positive gain the next month. These should be corrected.

- There are pathological cases where the scaling can cause one gage to have a large positive flow, while the adjacent gage has a large negative flow. This can occur when the two extended values are similar in magnitude but opposite in sign. These need to be adjusted.

- For this work, the changes are tracked in the file ***hand_mods.xls*****.**

To test that all flows, gains, and drainage areas have been properly accounted for, we check that the "reconstituted," unimpaired gage flows match the actual unimpaired gage flows. In this case, reconstituted means that the flow at a given gage is computed by summing the appropriate upstream flows and gains. File ***gage_comp.xls*** compares the actual monthly unimpaired gage flows with the computed values from ***flows_gains_month.dss***. To test that all the drainage areas have been properly accounted, file ***model_comp.xls*** compares the same actual unimpaired gage flows to the gage flows reconstituted from the computed OASIS inflows.

Negative inflow adjustments. First, monthly flows were adjusted in the file ***hand_mods.xls***. Revisions were done for Lure, Clif and Lawn flows and Boil and Gaff gains in the ***unimpaired.dss*** file, and for the same flows/gains in file ***scale_flow_gain.dss***,to adjust for negatives from the scaling process as noted above.

Second, to prevent model infeasibility from provisional inflows (see Appendix C), unrealistic releases from upstream reservoirs, or unrealistic water supply shortages, we added code in the OCL to filter remaining daily negative inflows. The negative inflow is "stored" until there is a sufficiently positive inflow to release the accumulated negative flows, thereby preserving mass over a multi-day period. Since the negative inflows are generally very small and infrequent, the filtering has negligible impact on being able to match the monthly unimpaired gage flow.

Section 6. Time of Travel / Flow Routing

To account for time of travel along the Broad River, flow routing has been incorporated into both the development of inflows, and the handling of flows within the model. The time of travel reaches consist of Lake Lure to the Boiling Springs gage, and the Boiling Springs and 1st Broad Lawndale gages to the Gaffney gage. Lag coefficients were determined by minimizing the number of negatives caused in the daily gains. For the Lake Lure to Boiling Springs reach, the lag used is one day. For the Boiling Springs and 1st Broad to Gaffney reach, the lags used are 60% of today's upstream flows and 40% of yesterday's upstream flows. Flow routing for the development of inflows is handled in the HEC-DSSVue script called ***routing.py***. Flow routing within the model is handled in the OCL file ***routing.ocl***. The routing equations used for each gain are as follows:

Boiling Springs gain = Boiling Springs gage flow - yesterday's Lake Lure inflow

Gaffney gain = Gaffney flow – [0.6 * (today's Boiling Springs gage flow + today's Lawndale gage flow) + 0.4 * (yesterday's Boiling Springs gage flow + yesterday's Lawndale gage flow)]

Note that the computation of gains may include subtracting the flows from tributaries in the reach (see Section 3), which are not routed.

The first step in the routing is to create a daily routed time series for each location: Lure, Boil, and Lawn.

For example, this table illustrates the routed and actual unimpaired Lure flows for five days in 2000:

Date	Lure actual flow (cfs)	Lure routed flow (cfs)
1/1/1930	223.4	235.0
1/2/1930	271.0	223.4
1/3/1930	310.8	271.0
1/4/1930	223.4	310.8
1/5/1930	56.3	223.4

Next, these daily routed timeseries are averaged monthly; the monthly flows are used to compute gains for the above mentioned reaches.

For example, in January 1930, the monthly routed flow at Lure is 206, and the flow at Boiling Springs is 1720 cfs. The gain at Boiling Springs is computed by subtracting the routed Lure flow (as well as the un-routed Clif flow of 309 cfs) from the Boiling Springs flow, or 1720 – 206 – 309 = 1205 cfs.

For inflow development, upstream flows are routed before being subtracted from the downstream flow to compute a gain. In the model runs, routing is handled using a

routing reservoir, which holds back upstream flows for the appropriate amount of time before releasing downstream.

Section 7. Extending the Record Beyond September 2009

As mentioned earlier, the finalized inflow record ends on September 30, 2009. This section describes how to finalize updates to the record when new records (including impairments) become available. This is not to be confused with *provisional* updates used to facilitate real-time forecasting, which are done directly from the model interface using the Update Record tab. Let us assume that we are adding data from October 1, 2009 to September 30, 2010. Note that we are only adding to the record. We are not changing any of the values prior to October 2009.

1. Assemble the new gage records in Table 4 below and place in the files ***gage_day.dss*** and ***gage_month.dss***.

2. Compute the impairments at each gage and add them to the gage flows. This has been done in the unimpairment spreadsheet described earlier (called ***inflow_unimpairment.xls)***. Next put the daily unimpaired flows into the spreadsheet *unimpaired_summary.xls* to compute the monthly unimpaired flows using a spreadsheet pivot table.

3. Append the new unimpaired flows to the ***unimpaired.dss*** file (monthly averages) and to the ***unimpaired_daily.dss*** file (daily data) where appropriate.

4. Copy the following files into the folder *c:\Program Files\HEC\HecDssVue* (assuming the DSSVue has been installed in the default location).

   fillin.cf
   fillin.exe
   gage_day.dss
   gage_month.dss
   path_list.dat
   unimpaired.dss
   unimpaired_daily.dss

5. Copy the following files in folder of *c:\Program Files\HEC\HecDssVue\HecDssVue\scripts.* If extending the record to a year beyond 2009, be sure to change the ending years in the script from either '09 or 2009 to the appropriate year. Use the script files from the folder "scripts_2010", in which this change has already been made.

   routing.py
   compute_gain.py
   convert_day_to_month.py
   convert_month_to_day.py
   disaggregate.py
   scale_flow_gain.py

6. Execute the script ***01_routing.py.*** This reads from file ***unimpaired_daily.dss***, creates routed time-series where necessary (see Section 6), and writes monthly routed flows to ***unimpaired.dss*** and ***extend_flow_gain.dss***.

7. Transfer the monthly flows from ***unimpaired.dss*** to *hand_mods.xls*. Here perform necessary modifications to remove negative gains from the records, following the examples of previous modifications. Then paste the modified flows and routed flows back into unimpaired.dss and ***extend_flow_gain.dss.***

8. Execute the script ***02_compute_gain.py.*** This reads from file ***unimpaired.dss***, computes all the gains, and writes to the files ***extend_flow_gain.dss*****,** ***scale_flow_gain.dss*****, and** ***fillin_input.dss.***

9. Close DSSVUE, and execute *fillin* by double-clicking it. This updates the file ***extend_flow_gain.dss*** with filled-in flows. Thus, this file is a combination of the actual unimpaired values and extended values.

10. Open up DSSVue again, execute the script ***03_scale_flow_gain.py***, which does the calculations described in Section 3. This reads from file ***extend_flow_gain.dss*** and updates file ***scale_flow_gain.dss***.

11. Import the appropriate flows and gains from ***scale_flow_gain.dss*** into *hand_mods.xls* and perform hand modifications to remove negative flows and gains. Paste modified flows/gains back into ***scale_flow_gain.dss***.

12. Execute the script ***04_convert_month_to_day.py***. This reads from files ***gage_month.dss*** and ***scale_flow_gain.dss*** to generate a daily record from the monthly records. The daily records are used to disaggregate the flows, as shown in Section 3. File ***month_day.dss*** is written.

13. Execute the script ***05_disaggregate.py*** to convert the monthly flows and gains at the gages to daily. File ***flow_gain.dss*** is produced; this file has the data that are read by OASIS.

14. Execute the script ***06_convert_day_to_month.py*** to convert the daily flows and gains at the gages into monthly values. The purpose of this step is to import the monthly flows and gains into ***gage_comp.xls*** to ensure that all the accounting has been properly done.

15. Once the correct accounting has been confirmed, place the ***flow_gains.dss*** file into the basedata folder, overwriting the original.

16. Execute the OASIS run called “Compute_Inflows”, beginning October 1, 2009 and ending September 30, 2010. This run contains a file called “set_inflows.ocl” that assigns inflows to the OASIS nodes.

Then go to the run folder, open the ***ouput.dss*** file, click View → Refresh Catalog. Select all of the inflow nodes (i.e, those which have a pathname labeled inflow) and convert from acre feet (af) to cubic feet per second (cfs) by doing the following: click Utilities → Math Functions, in Arithmetic tab select Divide from the pull down menu, click the Data Set radio button and highlight all of the inflow records, click the Constant radio button and enter 1.9835 into the field, and then press the Compute button at the bottom of the window. Once the computation has completed, close the Math Functions window and click Yes when prompted to save the changes. Now select all of the inflow records and click Edit → Tabular Edit, and change all of the units to cfs, and all of the units Type to PER-AVER. Close the editing window and save your changes.

Next select all of the records, and copy and append to the ***basedata.dss*** file in the basedata folder. The basedata file will now contain this finalized data from October 1, 2009 to September 30, 2010. This basedata file will be used for all future runs.

Table 4. Gages Needed to Extend the Record Beyond September 2009

USGS Number	Description	Period of Record	Ref. Name	Drainage Area
02149000	COVE CREEK NEAR LAKE LURE, NC	01/1951 - present	Cove	79
02150495	SECOND BROAD RIVER NEAR LOGAN, NC	10/1998 - present	Logn	86.2
02151500	BROAD RIVER NEAR BOILING SPRINGS, NC	07/1925 - present	Boil	875
02152100	FIRST BROAD RIVER NEAR CASAR, NC	03/1959 - present	Casr	60.5
02138500	LINVILLE RIVER NEAR NEBO, NC	07/1922 - present	Nebo	66.7
02143000	HENRY FORK NEAR HENRY RIVER, NC	08/1925 - present	Hnry	83.2
02143040	JACOB FORK AT RAMSEY, NC	10/1961 - present	Jacb	25.7
02143500	INDIAN CREEK NEAR LABORATORY, NC	09/1951 - present	Indn	69.2

APPENDIX C –

Provisional Inflow Data Development

The current methodology for developing model inflow data does not lend itself well to frequent updates that will be necessary for real-time position analysis. The current methodology requires a large amount of input gage data (using 6 gages); impairments from reservoir operations, water supply, wastewater returns, and agricultural withdrawals; correction to negative inflows that could otherwise cause model infeasibility; and scaling of gains to ensure that filled-in data for gages with missing data preserves the known volume of flow at downstream gages. Obtaining impairment data alone (which are necessary to unimpair the gage flows) is the most time-intensive part of the updating process.

HydroLogics has developed a simplified, *provisional* procedure that will enable weekly or monthly updates to be made, later overridden by periodic annual updates using the current methodology. It is meant to provide a representative inflow to key points in the basin. The downloading of data and calculations for the provisional update are handled automatically within the GUI.

To simplify the update as much as possible, the procedure eliminates the need for most impairments and the concern over negative inflows that could lead to model infeasibility. The assumption is that most of the net impairments (withdrawals less discharges) in the basin are small and occur within a reach and therefore are not likely to have much effect on the natural inflow.

Negative inflows can occur when the downstream gage flow is less than the upstream gage flow (which is usually due to time of travel issues). These only pose a modeling problem if there is not enough water in the river or reservoir to handle them, which is rare. As a precaution, when gains are negative, the model's OCL is used to filter them to maintain model feasibility.

The simplified procedure is detailed below in the following steps. As noted, all of the data acquisition and calculations are automatically done within the model. The user should do a manual QA/QC check on the downloaded data before updating the record.

Step 1: Obtain from the USGS web site the daily gage data (in cfs) for the following gages: Cove Creek near Lake Lure (Cove), 2nd Broad near Logan (Logn), Broad River at Boiling Springs (Boil), 1st Broad near Casar (Casr), 1st Broad near Lawndale (Lawn), and Indian Creek near Laboratory (Indn).

Step 2: Obtain rainfall data (in/day) from the Gaston Shoals COOP station.

Step 3: Compute the net evaporation for all reservoirs in the basin using the precipitation data collected above and a monthly evaporation pattern.

Step 4: Set the inflow to Lakes Summit, Adger and Lure using drainage area adjustments of the Cove gage.

Step 5: Set the 2nd Broad Forest City and Cliffside inflows using drainage area adjustments of the Logn gage.

Step 6: Compute the total Boiling Springs gain as the Boil gage flow minus the Summit, Adger and Lure inflows (lagged one day) and minus the 2nd Broad inflows. To better reflect the unimpaired inflow in this reach, an average of 15 cfs is added to this gain to account for upstream withdrawals and discharges (the most significant being the Cliffside Steam Plant withdrawal).

Step 7: Allocate the computed Boiling Springs gain to inflows for the Broad/Green confluence, the Cliffside dam, and the Boiling Springs gage.

Step 8: Set the inflow for the first Broad inflow locations using area adjustments of the Casr gage.

Step 9: Set the inflows for Gaston Shoals, Moss Lake, Buffalo Creek and Gaffney using area adjustments of the Indian Creek gage.

APPENDIX D –

Model Weighting Description

This report provides a detailed account of the weighting of nodes and arcs in the Broad River Basin model. Weighting greatly reduces the amount of coding required by the programmer, especially the conditional If → Then statements that are inherent in many software packages. OASIS operates using a linear program solver, which means that it tries to maximize the overall value of allocating water subject to the goals (which have associated weights) and constraints (which must be met). The general strategy with goal-setting is to assign weights to mimic the real-world operating goals. For example, setting a reservoir's storage weight higher than that of an unassociated demand downstream will prevent water from being released from that reservoir to meet the demand. Weighting is also used to properly dictate minimum releases and other flow targets.

Note that weighting is mostly relative. If the weight in storage (say 2) is higher than a weight for demand (say 1), the demand will not be met. Minimum flow weights are handled differently at times since they can be additive. If there are multiple minimum flow locations downstream of a reservoir, OASIS will assign value to the minimum releases based on the sum of those weights. So if there are three locations, each having a weight of 1, the model will get 3 points releasing water from an upstream reservoir to meet the minimum flows. If the storage weight is 2, then the reservoir will draw down to meet the minimum flows. Flow exceeding the minimum flow does not get any additional value, so excess water will stay in storage unless the reservoir is spilling. The user manual for OASIS provides more description on how model weighting works.

Each section of this document describes a portion of the model, progressing downstream.

Upper Broad River

The reservoirs on the upper Broad are set up to prevent water being released to meet unrelated needs further downstream.

The reservoir storage weights in this area are:

		Storage Zone Weights			
Reservoir	**Node Number**	**A**	**B**	**C**	**D**
Lake Summit	010	550	550	250	-10
Lake Adger	040	500	500	200	-10
Lake Lure	070	500	500	200	-10

Other weights in the area include:

Description	Node/arc Number	Weight
Lake Lure Min Release	070.080	225
Summit Ag (Agricultural Demand)	012	300
Adger Ag	042	225
Polk Co. Demand	061	215
Lure Ag	072	250
BRWA Demand	086	100
Broad/Green Ag	102	80
Logan Ag	152	80
Forest City Demand	186	80
2nd Cliffside Ag	192	80

Storage in Lake Summit, upstream of Lake Adger on the Green River, is weighted higher than Adger storage. The municipal water supply demands placed on Adger receive higher weight than usable storage in the reservoir (as defined by C zone weight). Dead storage, which is inaccessible storage, is represented by the A zone and receives a higher weight than the demand weight so that the model does not dip into this storage. The B Zone represents storage in between the lower rule and the dead storage zone. The flood pool is represented by the D zone, and has a negative weight to discourage storing water in this zone.

For Lake Lure the minimum release weight is higher than the usable storage weight in zone C.

Middle Broad River

The weighting for the Cliffside Dam is set up to prevent water from being released upstream to meet its needs and from releasing water to meet unrelated needs downstream.

The reservoir storage weights in this area are:

		Storage Zone Weights			
Reservoir	**Node Number**	**A**	**B**	**C**	**D**
Cliffside Dam	220	100	100	75	-10

Other weights in the area include:

Description	Node/arc Number	Weight
Cliffside Ag	222	80
Cliffside Demand	224	80
Boiling Springs Ag	252	25

The higher B and C zone weights for upstream reservoirs relative to the Cliffside Dam prevent releases from being made to supplement the reservoir (except when spilling, when flow cannot be controlled). From Cliffside Dam, the demand weights are higher so they take priority over storage. Since storage in the usable pool (zone C) gets last priority, this zone get the lowest weight. However, this weight is set higher than demands downstream (e.g., Boiling Spring agricultural use).

First Broad River

The reservoir storage weights in this area are:

		Storage Zone Weights			
Reservoir	**Node Number**	**A**	**B**	**C**	**D**
Stice Shoals	440	75	75	50	-10

Other weights in this area consist of the following:

Description	Node/arc Number	Weight
Shelby Min. Flow	420.440	75
Casar Ag	412	175
CCW Demand	416	80
Lawndale Ag	422	80
Shelby Demand	436	70
Stice Shoals Ag	442	80

The most upstream demands receive the highest weight since they are met first. The minimum flow-by requirement for Shelby is higher than the demand weight but lower than upstream demands, so the upstream users are not shorted in order to meet the requirement.

Lower Broad River (Including Buffalo Creek)

This area includes the Broad River between Boiling Springs and the confluence withBuffalo Creek.

The reservoir storage weights in this area are:

		Storage Zone Weights			
Reservoir	**Node Number**	**A**	**B**	**C**	**D**
Gaston Shoals	550	10	10	5	-10
Moss Lake	600	500	75	50	-10

Other weights in the area include:

Description	Node/arc Number	Weight
Gaston Shoals Bypass	550.555	8
Gaston Shoals Min Flow	550.700	7
Moss Lake Min Flow	600.610	80
Gaston Shoals Ag	552	11
Gaffney Demand	554	11
KM Ag	602	85
Kings Mt Demand	604	70
CNA Demand	645	25
Buffalo Ck Ag	652	25

Gaston Shoals has two minimum flow requirements-- the release and a bypass flow. While the bypass has higher priority than the release, staying within the normal pool (Zone C) takes priority over both. So first, the model will make sure there is enough inflow to maintain storage in zone C. Surplus inflow after that will go towards meeting the bypass flow, and any remaining available flow will go towards meeting the minimum release.

Moss Lake is set up to first meet a minimum release requirement, and then the local water supply demands.