Appendix J

Connecticut River Optimization Model

CROME model formulation & parameterization

The Connecticut River Optimization Modeling Environment (CROME) is formulated in the proprietary LINGO™ optimization software environment. LINGO™ provides a modeling environment where optimization problems are intuitively formulated using the software's set-based modeling language and solved using the software's suite of linear, binary, and nonlinear optimization algorithms (LINDO Systems 2010). The CROME model uses the simplex-based solver to solve the linear program (LP) hydropower optimization formulation and takes advantage of LINGO's™ interactive data management capabilities to import modeled input data from a Visual Basic for Applications (VBA) enabled Microsoft Excel workbook. The following sections explain the CROME modeling environment, the components of the LP formulation, the scripts used to model the Baseline and Run-of-River scenarios, and important model parameters.

General model structure

The major components of the CROME modeling environment include model inputs, modeling procedure, and post-processing. Figure 1 shows the general structure of the CROME modeling framework, including flow inputs to the coupled Excel-LINGO™ model and post processing in the open-source R coding language. Model inputs are housed within the large spreadsheet which contains CRUISE-estimated daily flows, other historical data like energy prices, and documented physical and operating parameters used to constrain the multiple objective model. Using a VBA script, these data are passed to the LINGO™ model, and a reservoir operations optimization is executed at yearly increments. Modeled output is aggregated in simple text files for post-processing in the R coding environment where necessary data adjustments can be made before performing model analyses.

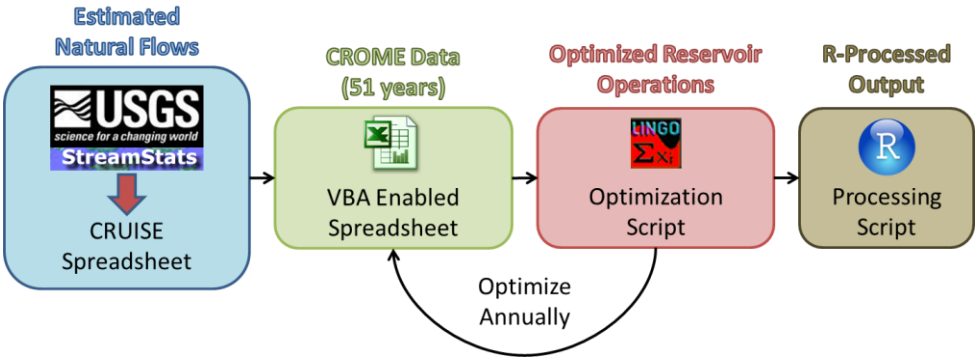

Figure 1 - Schematic of the generalized CROME workflow

LP formulation

The following section defines the unique components of the linear program formulation for CROME including the objective function formulation and operational constraints used to limit hydropower operations to real-world operations in the order presented in the LINGO™ script defined in the next section.

Objective function

The objective function is a mathematical representation of the operation objectives used to govern a set of dam operations. In the objective function, these operational objectives are stated such that the model is penalized for deviating from an operational target and the optimization algorithm is tasked with minimizing the aggregate of the penalty-based formulation.

Figure 2 provides a schematic outlining a simple example which demonstrates a multiple objective formulation for a single reservoir. Here, the objective function incorporates penalties related to reservoir storage level, ecological flows, and flood flows which correspond to unique operational targets. For instance, the operational target for ecological flows may be some arbitrary flow rate defined on the x axis in blue. The penalty for deviating above or below this flow rate is shown on the y axis, where the area under the curve between the target and actual flow rate defines the ecological flow penalty. The model seeks to minimize the area under each of the three penalty curves so that the sum of these three is optimally minimized.

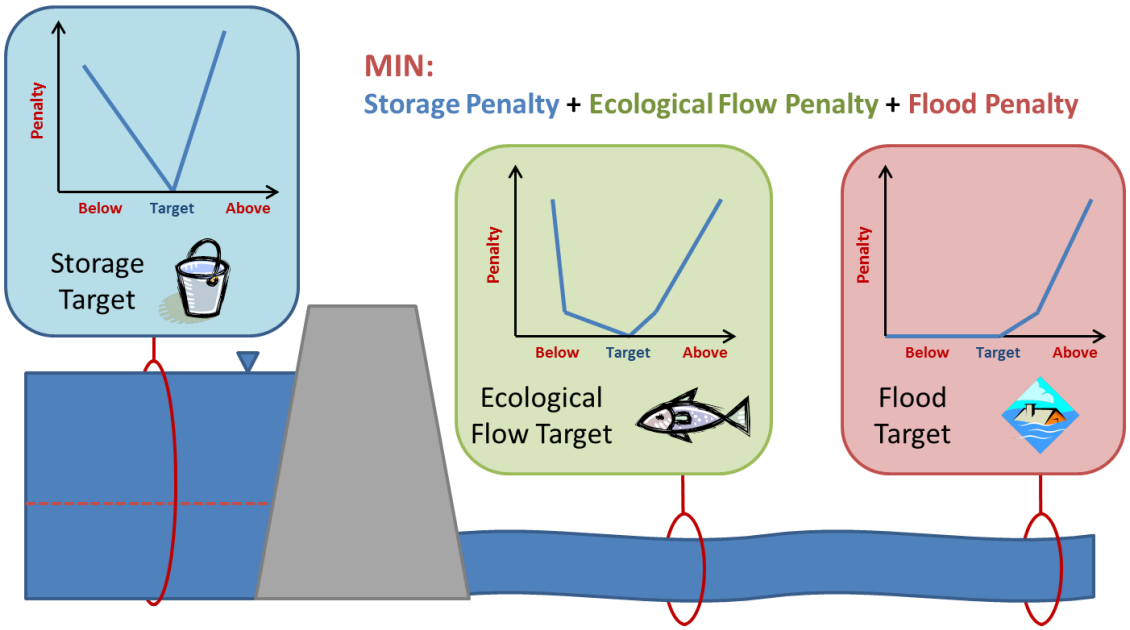

Figure 2 - Example single reservoir optimization schematic.

By varying the relative weighting in each of the objective function penalties, CROME is capable of comparing various operating regimes. In the example presented in Figure 2, the objective function penalty for flood control could be mathematically weighted to be of greater importance than the ecological flow penalty, vice versa, and at increments in between. This approach allows CROME to create a suite of model output which demonstrates the flow regime impacts if operational goals were to change from only flood control, to only ecological flows or some more likely combination of the two.

While only one reservoir and the three operating objectives were used in this example, the model is formulated to address various other objectives including hydropower production, water supply, and recreation at a total of 54 dams.

Ecological Objectives

The ecological objectives defined in CROME are slightly more complex than the simple flood control or storage target penalties that are governed by predefined dam operational data. These objectives are the product of data gathered from a 2-day workshop where ecologists identified and defined the importance that flow magnitudes and timings have on various species including macroinvertibrates, mussels, resident fish, diadromous fish, and floodplains.

These relationships inform penalty functions like the one shown in Figure 2 which vary at different flow magnitudes and months in order to demonstrate the importance of a certain type of flow event during a certain time of year.

For instance, an ecological penalty for floodplains seeks to ensure high flows in the springtime by increasing the slope of the penalty function around high flow periods. This encourages the model to choose operations which provide ecologically necessary periods of high flows for floodplain tree growth. A similar approach is applied to low flows during summer months for resident fish which need sustained low flow periods to encourage population growth among juveniles.

The importance of each species is spatially defined at various 'Eco-Node' locations chosen across the Connecticut River basin to represent points of concern for the previously discussed species.

Model constraints

To ensure that CROME accurately mimics operations on the current reservoir system, various physical and operational constraints are applied to the model to define the basic functional components of unique dam facilities and their operation.

Continuity

A mass balance is applied at each modeled reservoir to ensure continuity of flows and water storage along the mainstem. The mass balance constraint simply states that the storage of any given reservoir is the summation of its storage at the previous time step and its inflows and releases at the current time step. To ensure continuity of reservoir storage across modeled years, the initial and final storages are constrained to the same value for each year.

Physical and operating constraints

Each facility is constrained to its physical limitations including useable reservoir storage, maximum turbine flow rate, and maximum flow rates. Operating constraints are applied in the form of minimum flows and release ramping rates. The ramping rates are either defined by reservoir documentation, or applied to ensure realistic modeled output.

Full Basin LINGO script

The following section shows the LINGO™ script code for the objective and constraint formulation of the Full Basin model, designed to mimic real-world reservoir operations for the system of major dams in the greater Connecticut River Basin. Both the programming language and the variable nomenclature follow an intuitive scheme which is supported by commented explanations. Variables names follow the general format: *WSHD*_TYPE_NAME_VARIABLE where the WHSD is the optional three letter abbreviation for the watershed such as ASH for Ashuelot; TYPE is most generally RES for reservoir, G for flood gage, or EN for eco-node; the LOCATION is some character set such as WLD for Wilder Dam or NWAL for North Walpole flood gage, and VARIABLE may be Q for modeled inflow, R for modeled release, or ST for storage volume.

Each section defines the optimization model as it is scripted in LINGO for the daily, Full-Basin CROME model. Each subsection has a brief description explaining both the functionality of the code and its mathematical formulation.

```
MODEL:
!---Code for input data removed for brevity---;
```

Objective Function

This section defines the objective function as it is scripted in LINGO for the daily, Full-Basin CROME model. Three subsections exist, outlining the three major objectives: Eco-Targets (ecological flows), Flooding Targets (flood control), and Reservoir Operating Targets (storage targets & hydropower income generation). The objective function calculates the summation of each line of code across the three major objectives, individually minimizing penalties at each defined location with weight defined by the user's relative penalty weightings. These relative penalty weightings allow the user to define the importance of objectives such that differences in operations under ecological or flood control based scenarios may be explored.

```
!---Objective Function---;
MIN =
```

Eco-Targets

Here ecological targets are defined by the penalties accrued by deviating from estimated natural flow rates over the modeled period.

For instance, penalties for Eco-Node 1 are calculated by multiplying the weight (EN_1_WT) by the calculated level of flow deviation above or below the estimated natural flow rate at day I. Penalties calculated within the second tier of deviations (LOW/HIGH2) are more heavily weighted (multiplied by S2, a factor usually set at 2) than those in the first tier (LOW/HIGH1).

The yearly summation of these daily deviations (*@SUM(day(I):*) from estimated natural flow at Eco-Node 1 are aggregated with all remaining eco nodes which are calculated in the same format. The weight terms associated with each of these nodes are calculated relative to their contributing drainage area to establish a volumetrically-based relative importance.

```
!Eco-Targets;
EN_1_WT*@SUM(day(I): MAIN_EN_1_LOW1_BELOW(I) + MAIN_EN_1_HIGH1_ABOVE(I) +
S2*MAIN_EN_1_LOW2_BELOW(I) +  S2*MAIN_EN_1_HIGH2_ABOVE(I)) +
EN_6_WT*@SUM(day(I): MAIN_EN_6_LOW1_BELOW(I) + MAIN_EN_6_HIGH1_ABOVE(I) +
S2*MAIN_EN_6_LOW2_BELOW(I) +  S2*MAIN_EN_6_HIGH2_ABOVE(I)) +
EN_9_WT*@SUM(day(I): MAIN_EN_9_LOW1_BELOW(I) + MAIN_EN_9_HIGH1_ABOVE(I) +
S2*MAIN_EN_9_LOW2_BELOW(I) +  S2*MAIN_EN_9_HIGH2_ABOVE(I)) +
EN_11_WT*@SUM(day(I): MAIN_EN_11_LOW1_BELOW(I) + MAIN_EN_11_HIGH1_ABOVE(I) +
S2*MAIN_EN_11_LOW2_BELOW(I) + S2*MAIN_EN_11_HIGH2_ABOVE(I)) +
EN_19_WT*@SUM(day(I): MAS_EN_19_LOW1_BELOW(I) + MAS_EN_19_HIGH1_ABOVE(I) +
S2*MAS_EN_19_LOW2_BELOW(I) + S2*MAS_EN_19_HIGH2_ABOVE(I)) +
EN_21_WT*@SUM(day(I): BLK_EN_21_LOW1_BELOW(I) + BLK_EN_21_HIGH1_ABOVE(I) +
S2*BLK_EN_21_LOW2_BELOW(I) + S2*BLK_EN_21_HIGH2_ABOVE(I)) +
EN_22_WT*@SUM(day(I): SGR_EN_22_LOW1_BELOW(I) + SGR_EN_22_HIGH1_ABOVE(I) +
S2*SGR_EN_22_LOW2_BELOW(I) + S2*SGR_EN_22_HIGH2_ABOVE(I)) +
EN_24_WT*@SUM(day(I): MAIN_EN_24_LOW1_BELOW(I) + MAIN_EN_24_HIGH1_ABOVE(I) +
S2*MAIN_EN_24_LOW2_BELOW(I) +  S2*MAIN_EN_24_HIGH2_ABOVE(I)) +
EN_26_WT*@SUM(day(I): MAIN_EN_26_LOW1_BELOW(I) + MAIN_EN_26_HIGH1_ABOVE(I) +
S2*MAIN_EN_26_LOW2_BELOW(I) +  S2*MAIN_EN_26_HIGH2_ABOVE(I)) +
EN_33_WT*@SUM(day(I): WST_EN_33_LOW1_BELOW(I) + WST_EN_33_HIGH1_ABOVE(I) +
S2*WST_EN_33_LOW2_BELOW(I) +  S2*WST_EN_33_HIGH2_ABOVE(I)) +
EN_37_WT*@SUM(day(I): ASH_EN_37_LOW1_BELOW(I) + ASH_EN_37_HIGH1_ABOVE(I) +
S2*ASH_EN_37_LOW2_BELOW(I) +  S2*ASH_EN_37_HIGH2_ABOVE(I)) +
EN_39_WT*@SUM(day(I): MLR_EN_39_LOW1_BELOW(I) + MLR_EN_39_HIGH1_ABOVE(I) +
S2*MLR_EN_39_LOW2_BELOW(I) +  S2*MLR_EN_39_HIGH2_ABOVE(I)) +
EN_45_WT*@SUM(day(I): DRF_EN_45_LOW1_BELOW(I) + DRF_EN_45_HIGH1_ABOVE(I) +
S2*DRF_EN_45_LOW2_BELOW(I) +  S2*DRF_EN_45_HIGH2_ABOVE(I)) +
EN_59_WT*@SUM(day(I): FAR_EN_59_LOW1_BELOW(I) + FAR_EN_59_HIGH1_ABOVE(I) +
S2*FAR_EN_59_LOW2_BELOW(I) +  S2*FAR_EN_59_HIGH2_ABOVE(I)) +
EN_74_WT*@SUM(day(I): MAIN_EN_74_LOW1_BELOW(I) + MAIN_EN_74_HIGH1_ABOVE(I) +
S2*MAIN_EN_74_LOW2_BELOW(I) +  S2*MAIN_EN_74_HIGH2_ABOVE(I)) +
EN_76_WT*@SUM(day(I): WSF_EN_76_LOW1_BELOW(I) + WSF_EN_76_HIGH1_ABOVE(I) +
S2*WSF_EN_76_LOW2_BELOW(I) +  S2*WSF_EN_76_HIGH2_ABOVE(I)) +
EN_78_WT*@SUM(day(I): WSF_EN_78_LOW1_BELOW(I) + WSF_EN_78_HIGH1_ABOVE(I) +
S2*WSF_EN_78_LOW2_BELOW(I) +  S2*WSF_EN_78_HIGH2_ABOVE(I)) +
EN_80_WT*@SUM(day(I): CHP_EN_80_LOW1_BELOW(I) + CHP_EN_80_HIGH1_ABOVE(I) +
S2*CHP_EN_80_LOW2_BELOW(I) +  S2*CHP_EN_80_HIGH2_ABOVE(I)) +
EN_88_WT*@SUM(day(I): MAIN_EN_88_LOW1_BELOW(I) + MAIN_EN_88_HIGH1_ABOVE(I) +
S2*MAIN_EN_88_LOW2_BELOW(I) +  S2*MAIN_EN_88_HIGH2_ABOVE(I)) +
EN_91_WT*@SUM(day(I): ASH_EN_91_LOW1_BELOW(I) + ASH_EN_91_HIGH1_ABOVE(I) +
S2*ASH_EN_91_LOW2_BELOW(I) +  S2*ASH_EN_91_HIGH2_ABOVE(I)) +
EN_95_WT*@SUM(day(I): WST_EN_95_LOW1_BELOW(I) + WST_EN_95_HIGH1_ABOVE(I) +
S2*WST_EN_95_LOW2_BELOW(I) +  S2*WST_EN_95_HIGH2_ABOVE(I)) +
```

```
EN_98_WT*@SUM(day(I): MAIN_EN_98_LOW1_BELOW(I) + MAIN_EN_98_HIGH1_ABOVE(I) +
S2*MAIN_EN_98_LOW2_BELOW(I) +  S2*MAIN_EN_98_HIGH2_ABOVE(I)) +
EN_100_WT*@SUM(day(I): MAIN_EN_100_LOW1_BELOW(I) + MAIN_EN_100_HIGH1_ABOVE(I) +
S2*MAIN_EN_100_LOW2_BELOW(I) +  S2*MAIN_EN_100_HIGH2_ABOVE(I)) +
EN_136_WT*@SUM(day(I): DRF_EN_136_LOW1_BELOW(I) + DRF_EN_136_HIGH1_ABOVE(I) +
S2*DRF_EN_136_LOW2_BELOW(I) + S2*DRF_EN_136_HIGH2_ABOVE(I)) +
EN_137_WT*@SUM(day(I): DRF_EN_137_LOW1_BELOW(I) + DRF_EN_137_HIGH1_ABOVE(I) +
S2*DRF_EN_137_LOW2_BELOW(I) + S2*DRF_EN_137_HIGH2_ABOVE(I)) +
EN_139_WT*@SUM(day(I): FAR_EN_139_LOW1_BELOW(I) + FAR_EN_139_HIGH1_ABOVE(I) +
S2*FAR_EN_139_LOW2_BELOW(I) + S2*FAR_EN_139_HIGH2_ABOVE(I)) +
EN_149_WT*@SUM(day(I): FAR_EN_149_LOW1_BELOW(I) + FAR_EN_149_HIGH1_ABOVE(I) +
S2*FAR_EN_149_LOW2_BELOW(I) + S2*FAR_EN_149_HIGH2_ABOVE(I)) +
EN_152_WT*@SUM(day(I): MAIN_EN_152_LOW1_BELOW(I) + MAIN_EN_152_HIGH1_ABOVE(I) +
S2*MAIN_EN_152_LOW2_BELOW(I) + S2*MAIN_EN_152_HIGH2_ABOVE(I)) +
```

Flooding Targets

Here flooding targets are defined by the penalties accrued by exceeding pre-defined flood levels at each of the real world, or proxy flood gages in the basin.

For instance, penalties for flooding on the Indian River are calculated by multiplying the weight (IND_G_IND_TARG_WT) by the calculated level of flow deviation above the flow definition for either warning flow (*TARG_ABOVE) or flood flow (*TARG2_ABOVE) levels associated with the flood gage. Penalties calculated for the flood flow level are more heavily weighted than those in the warning flow levels to encourage the model to prevent high flood flows.

The yearly summation of these daily flooding levels (*@SUM(day(I):*) from the Indian River gage are aggregated with all remaining flood gage nodes which are calculated in the same format. The weight terms associated with each of these nodes are calculated relative to their contributing drainage area to establish a volumetrically-based relative importance.

```
!Flooding;
IND_G_IND_TARG_WT * @SUM(day(I): IND_G_IND_TARG_ABOVE(I)) +     !Indian Gauge at
Indian Gauge;
MAIN_G_WLEB_TARG_WT * @SUM(day(I): MAIN_G_WLEB_TARG_ABOVE(I)) +   !West Lebanon Gauge;
MAIN_G_NWAL_TARG_WT * @SUM(day(I): MAIN_G_NWAL_TARG_ABOVE(I)) +   !North Walpole
Gauge;
MAIN_G_MONT_TARG_WT * @SUM(day(I): MAIN_G_MONT_TARG_ABOVE(I)) +   !Montague City
Gauge;
ASH_G_HIN_TARG_WT * @SUM(day(I): ASH_G_HIN_TARG_ABOVE(I)) +   !Hinsdale Gauge;
MLR_G_ERV_TARG_WT * @SUM(day(I): MLR_G_ERV_TARG_ABOVE(I)) +   !Erving Gauge;
CHP_G_GIB_TARG_WT * @SUM(day(I): CHP_G_GIB_TARG_ABOVE(I)) +   !Gibbs Crossing Gauge;
CHP_G_IND_TARG_WT * @SUM(day(I): CHP_G_IND_TARG_ABOVE(I)) +   !Indian Gauge;
WSF_G_WSF_TARG_WT * @SUM(day(I): WSF_G_WSF_TARG_ABOVE(I)) +   !Westfield Gauge;
FAR_G_RIV_TARG_WT * @SUM(day(I): FAR_G_RIV_TARG_ABOVE(I)) +   !Unionville in Riverton;
FAR_G_UNV_TARG_WT * @SUM(day(I): FAR_G_UNV_TARG_ABOVE(I)) +   !Unionville in
Farmington;
MAIN_G_THOM_TARG_WT * @SUM(day(I): MAIN_G_THOM_TARG_ABOVE(I)) +   !Thompsonville;
MAIN_G_HART_TARG_WT * @SUM(day(I): MAIN_G_HART_TARG_ABOVE(I)) +   !Hartford;

IND_G_IND_TARG2_WT * @SUM(day(I): IND_G_IND_TARG2_ABOVE(I)) +     !Indian Gauge at
Indian Gauge;
MAIN_G_WLEB_TARG2_WT * @SUM(day(I): MAIN_G_WLEB_TARG2_ABOVE(I)) +   !West Lebanon
Gauge;
MAIN_G_NWAL_TARG2_WT * @SUM(day(I): MAIN_G_NWAL_TARG2_ABOVE(I)) +   !North Walpole
Gauge;
```

```
MAIN_G_MONT_TARG2_WT * @SUM(day(I): MAIN_G_MONT_TARG2_ABOVE(I)) +   !Montague City
Gauge;
ASH_G_HIN_TARG2_WT * @SUM(day(I): ASH_G_HIN_TARG2_ABOVE(I)) +   !Erving Gauge;
MLR_G_ERV_TARG2_WT * @SUM(day(I): MLR_G_ERV_TARG2_ABOVE(I)) +   !Erving Gauge;
CHP_G_GIB_TARG2_WT * @SUM(day(I): CHP_G_GIB_TARG2_ABOVE(I)) +   !Gibbs Crossing Gauge;
CHP_G_IND_TARG2_WT * @SUM(day(I): CHP_G_IND_TARG2_ABOVE(I)) +   !Indian Gauge;
WSF_G_WSF_TARG2_WT * @SUM(day(I): WSF_G_WSF_TARG2_ABOVE(I)) +   !Westfield Gauge;
FAR_G_RIV_TARG2_WT * @SUM(day(I): FAR_G_RIV_TARG2_ABOVE(I)) +   !Unionville in
Riverton;
FAR_G_UNV_TARG2_WT * @SUM(day(I): FAR_G_UNV_TARG2_ABOVE(I)) +   !Unionville in
Farmington;
MAIN_G_THOM_TARG2_WT * @SUM(day(I): MAIN_G_THOM_TARG2_ABOVE(I)) +   !Thompsonville;
MAIN_G_HART_TARG2_WT * @SUM(day(I): MAIN_G_HART_TARG2_ABOVE(I)) +   !Hartford;

OMP_RES_UNV_ST_FLOOD_TARG_WT * @SUM(day(I): OMP_RES_UNV_ST_FLOOD_TARG_ABOVE(I)) +
OTT_RES_NHD_ST_FLOOD_TARG_WT * @SUM(day(I): OTT_RES_NHD_ST_FLOOD_TARG_ABOVE(I)) +
BLK_RES_NSP_ST_FLOOD_TARG_WT * @SUM(day(I): BLK_RES_NSP_ST_FLOOD_TARG_ABOVE(I)) +
WST_RES_BMD_ST_FLOOD_TARG_WT * @SUM(day(I): WST_RES_BMD_ST_FLOOD_TARG_ABOVE(I)) +
WST_RES_TWN_ST_FLOOD_TARG_WT * @SUM(day(I): WST_RES_TWN_ST_FLOOD_TARG_ABOVE(I)) +
ASH_RES_SMD_ST_FLOOD_TARG_WT * @SUM(day(I): ASH_RES_SMD_ST_FLOOD_TARG_ABOVE(I)) +
ASH_RES_OBD_ST_FLOOD_TARG_WT * @SUM(day(I): ASH_RES_OBD_ST_FLOOD_TARG_ABOVE(I)) +
MLR_RES_TUL_ST_FLOOD_TARG_WT * @SUM(day(I): MLR_RES_TUL_ST_FLOOD_TARG_ABOVE(I)) +
MLR_RES_BIR_ST_FLOOD_TARG_WT * @SUM(day(I): MLR_RES_BIR_ST_FLOOD_TARG_ABOVE(I)) +
WSF_RES_KVL_ST_FLOOD_TARG_WT * @SUM(day(I): WSF_RES_KVL_ST_FLOOD_TARG_ABOVE(I)) +
WSF_RES_LVL_ST_FLOOD_TARG_WT * @SUM(day(I): WSF_RES_LVL_ST_FLOOD_TARG_ABOVE(I)) +
CHP_RES_BFD_ST_FLOOD_TARG_WT * @SUM(day(I): CHP_RES_BFD_ST_FLOOD_TARG_ABOVE(I)) +
CHP_RES_CBD_ST_FLOOD_TARG_WT * @SUM(day(I): CHP_RES_CBD_ST_FLOOD_TARG_ABOVE(I)) +
!FAR_RES_COL_ST_FLOOD_TARG_WT * @SUM(day(I): FAR_RES_COL_ST_FLOOD_TARG_ABOVE(I)) +
```

Reservoir Operation Targets

Here reservoir operation targets are defined by the penalties accrued for deviating from seasonal reservoir storage target levels, or the secondary objective to maximize daily hydropower revenues at the modeled hydropower facilities.

For instance, penalties for deviating from the USACE-defined seasonal target reservoir storage levels for the Black River's North Springfield flood control dam (BLK_RES_NSP_*) are calculated by the level of deviation above (ST_TARG_ABOVE) or below (ST_TARG_BELOW) the target. The storage objectives are included in the objective function to encourage reservoir storage operations that mimic those performed by real world operators following similar storage target guidelines.

Hydropower income is calculated by watershed. For instance, the three upper Connecticut hydropower facilities that comprise the 15-Mile Falls project are combined into one hydropower income term (UP3_TOTAL_INCOME_*). Note that the objective terms for hydropower revenues are preceded with a – sign because the general objective statement is to minimize penalties. The addition of a – sign forces the model to minimize the negative of, or maximize, modeled hydropower income.

```
!Upper Third;
UCN_RES_SCL_ST_TARG_WT * @SUM(day(I): UCN_RES_SCL_ST_TARG_ABOVE(I) +
UCN_RES_SCL_ST_TARG_BELOW(I)) + !Second Connecticut Lake Storage;
UCN_RES_FCL_ST_TARG_WT * @SUM(day(I): UCN_RES_FCL_ST_TARG_ABOVE(I) +
UCN_RES_FCL_ST_TARG_BELOW(I))+ !First Connecticut Lake Storage;
```

```
UCN_RES_LFR_ST_TARG_WT * @SUM(day(I): UCN_RES_LFR_ST_TARG_ABOVE(I) +
UCN_RES_LFR_ST_TARG_BELOW(I))+ !Lake Francis Storage;
MAIN_RES_MOR_ST_TARG_WT * @SUM(day(I):  MAIN_RES_MOR_ST_TARG_ABOVE(I) +
MAIN_RES_MOR_ST_TARG_BELOW(I))+ !Moore Storage Target;
MAIN_RES_COM_ST_TARG_WT * @SUM(day(I):  MAIN_RES_COM_ST_TARG_ABOVE(I) +
MAIN_RES_COM_ST_TARG_BELOW(I))+ !Moore Storage Target;
MAIN_RES_MCD_ST_TARG_WT * @SUM(day(I):  MAIN_RES_MCD_ST_TARG_ABOVE(I) +
MAIN_RES_MCD_ST_TARG_BELOW(I))- !Moore Storage Target;
UP3_TOTAL_INCOME_WT * @SUM(day(I): UP3_TOTAL_INCOME(I)) + !Total Income;

!Middle Third;
BLK_RES_NSP_ST_TARG_ABOVE_WT * @SUM(day(I): BLK_RES_NSP_ST_TARG_ABOVE(I)) + !North
Springfield Storage Above;
BLK_RES_NSP_ST_TARG_BELOW_WT * @SUM(day(I): BLK_RES_NSP_ST_TARG_BELOW(I)) + !North
Springfield Storage Below;
MAS_RES_GOO_ST_TARG_WT * @SUM(day(I): MAS_RES_GOO_ST_TARG_ABOVE(I) +
MAS_RES_GOO_ST_TARG_BELOW(I)) + !Goose Pond Storage;
MAS_RES_CRY_ST_TARG_WT * @SUM(day(I): MAS_RES_CRY_ST_TARG_ABOVE(I) +
MAS_RES_CRY_ST_TARG_BELOW(I)) + !Crystal Lake Storage;
MAS_RES_GRF_ST_TARG_WT * @SUM(day(I): MAS_RES_GRF_ST_TARG_ABOVE(I) +
MAS_RES_GRF_ST_TARG_BELOW(I)) + !Grafton Pond Storage;
MAS_RES_MSL_ST_TARG_WT * @SUM(day(I): MAS_RES_MSL_ST_TARG_ABOVE(I) +
MAS_RES_MSL_ST_TARG_BELOW(I)) + !Mascoma Lake Storage;
OMP_RES_UNV_ST_TARG_ABOVE_WT * @SUM(day(I): OMP_RES_UNV_ST_TARG_ABOVE(I)) + !Union
Village Storage Above;
OMP_RES_UNV_ST_TARG_BELOW_WT * @SUM(day(I): OMP_RES_UNV_ST_TARG_BELOW(I)) + !Union
Village Storage Below;
OTT_RES_NHD_ST_TARG_ABOVE_WT * @SUM(day(I): OTT_RES_NHD_ST_TARG_ABOVE(I)) + !North
Hartland Storage Above;
OTT_RES_NHD_ST_TARG_BELOW_WT * @SUM(day(I): OTT_RES_NHD_ST_TARG_BELOW(I)) + !North
Hartland Storage Below;
SUG_RES_LSU_ST_TARG_WT * @SUM(day(I): SUG_RES_LSU_ST_TARG_ABOVE(I) +
SUG_RES_LSU_ST_TARG_BELOW(I)) + !Lake Sunapee Storage;
MAIN_RES_BFA_ST_TARG_WT * @SUM(day(I): MAIN_RES_BFA_ST_TARG_ABOVE(I) +
MAIN_RES_BFA_ST_TARG_BELOW(I)) + !Bellows Falls Storage Target;
MAIN_RES_WLD_ST_TARG_WT * @SUM(day(I): MAIN_RES_WLD_ST_TARG_ABOVE(I) +
MAIN_RES_WLD_ST_TARG_BELOW(I)) - !Wilder Storage Target;
MID3_TOTAL_INCOME_WT * @SUM(day(I): MID3_TOTAL_INCOME(I)) +

!West;
WST_RES_BMD_ST_TARG_ABOVE_WT * @SUM(day(I): WST_RES_BMD_ST_TARG_ABOVE(I)) +
WST_RES_BMD_ST_TARG_BELOW_WT * @SUM(day(I): WST_RES_BMD_ST_TARG_BELOW(I)) +
WST_RES_TWN_ST_TARG_ABOVE_WT  * @SUM(day(I): WST_RES_TWN_ST_TARG_ABOVE(I)) +
WST_RES_TWN_ST_TARG_BELOW_WT  * @SUM(day(I): WST_RES_TWN_ST_TARG_BELOW(I)) +

!Ashuelot;
ASH_RES_SMD_ST_TARG_ABOVE_WT * @SUM(day(I): ASH_RES_SMD_ST_TARG_ABOVE(I)) +
ASH_RES_SMD_ST_TARG_BELOW_WT * @SUM(day(I): ASH_RES_SMD_ST_TARG_BELOW(I)) +
ASH_RES_OBD_ST_TARG_ABOVE_WT  * @SUM(day(I): ASH_RES_OBD_ST_TARG_ABOVE(I)) +
ASH_RES_OBD_ST_TARG_BELOW_WT  * @SUM(day(I): ASH_RES_OBD_ST_TARG_BELOW(I)) +

!Millers;
MLR_RES_MON_ST_TARG_WT * @SUM(day(I): MLR_RES_MON_ST_TARG_ABOVE(I)) +
MLR_RES_NEK_ST_TARG_WT * @SUM(day(I): MLR_RES_NEK_ST_TARG_ABOVE(I)) +
MLR_RES_BIR_ST_TARG_ABOVE_WT * @SUM(day(I): MLR_RES_BIR_ST_TARG_ABOVE(I)) +
MLR_RES_BIR_ST_TARG_BELOW_WT * @SUM(day(I): MLR_RES_BIR_ST_TARG_BELOW(I)) +
MLR_RES_TUL_ST_TARG_ABOVE_WT * @SUM(day(I): MLR_RES_TUL_ST_TARG_ABOVE(I)) +
MLR_RES_TUL_ST_TARG_BELOW_WT * @SUM(day(I): MLR_RES_TUL_ST_TARG_BELOW(I)) +

!Deerfield;
DRF_RES_SOM_ST_TARG_WT * @SUM(day(I): DRF_RES_SOM_ST_TARG_ABOVE(I) +
DRF_RES_SOM_ST_TARG_BELOW(I)) +
```

```
DRF_RES_HAR_ST_TARG_WT * @SUM(day(I): DRF_RES_HAR_ST_TARG_ABOVE(I) +
DRF_RES_HAR_ST_TARG_BELOW(I)) +
DRF_RES_SBG_ST_TARG_WT * @SUM(day(I): DRF_RES_SBG_ST_TARG_ABOVE(I) +
DRF_RES_SBG_ST_TARG_BELOW(I)) +
DRF_RES_SHR_ST_TARG_WT * @SUM(day(I): DRF_RES_SHR_ST_TARG_ABOVE(I) +
DRF_RES_SHR_ST_TARG_BELOW(I)) +
DRF_RES_FBR_ST_TARG_WT * @SUM(day(I): DRF_RES_FBR_ST_TARG_ABOVE(I) +
DRF_RES_FBR_ST_TARG_BELOW(I)) +
DRF_RES_DV5_ST_TARG_WT * @SUM(day(I): DRF_RES_DV5_ST_TARG_ABOVE(I) +
DRF_RES_DV5_ST_TARG_BELOW(I)) +
DRF_RES_DV4_ST_TARG_WT * @SUM(day(I): DRF_RES_DV4_ST_TARG_ABOVE(I) +
DRF_RES_DV4_ST_TARG_BELOW(I)) +
DRF_RES_DV3_ST_TARG_WT * @SUM(day(I): DRF_RES_DV3_ST_TARG_ABOVE(I) +
DRF_RES_DV3_ST_TARG_BELOW(I)) +
DRF_RES_GRD_ST_TARG_WT * @SUM(day(I): DRF_RES_GRD_ST_TARG_ABOVE(I) +
DRF_RES_GRD_ST_TARG_BELOW(I)) +
DRF_RES_DV2_ST_TARG_WT * @SUM(day(I): DRF_RES_DV2_ST_TARG_ABOVE(I) +
DRF_RES_DV2_ST_TARG_BELOW(I)) -
DRF_TOTAL_INCOME_WT * @SUM(day(I): DRF_TOTAL_INCOME(I)) +


!Chicopee;
CHP_RES_BFD_ST_TARG_ABOVE_WT*@SUM(day(I):CHP_RES_BFD_ST_TARG_ABOVE(I)) +
CHP_RES_BFD_ST_TARG_BELOW_WT*@SUM(day(I):CHP_RES_BFD_ST_TARG_BELOW(I)) +
CHP_RES_CBD_ST_TARG_ABOVE_WT*@SUM(day(I):CHP_RES_CBD_ST_TARG_ABOVE(I)) +
CHP_RES_CBD_ST_TARG_BELOW_WT*@SUM(day(I):CHP_RES_CBD_ST_TARG_BELOW(I)) +
CHP_RES_QWD_ST_TARG_WT*@SUM(day(I):CHP_RES_QWD_ST_TARG_BELOW(I)) +    !Quabbin Storage
Below Target 1;
CHP_RES_MMBR_ST_TARG_WT*@SUM(day(I):CHP_RES_MMBR_ST_TARG_BELOW(I)) +  !Mare
Meadow/Bickford Storage Below Target;
CHP_RES_RBD_ST_TARG_WT*@SUM(day(I): CHP_RES_RBD_ST_TARG_BELOW(I)) +  !Mare
Meadow/Bickford Storage Below Target;
CHP_RES_QWD_R_TARG_MAX_WT*@SUM(day(I):CHP_RES_QWD_R_TARG_MAX_ABOVE(I)) -     !Quabbin
Maximum Release;
CHP_TOTAL_INCOME_WT * @SUM(day(I): CHP_TOTAL_INCOME(I)) +

!Westfield;
WSF_RES_KVL_ST_TARG_ABOVE_WT*@SUM(day(I): WSF_RES_KVL_ST_TARG_ABOVE(I))+
WSF_RES_KVL_ST_TARG_BELOW_WT*@SUM(day(I): WSF_RES_KVL_ST_TARG_BELOW(I))+
WSF_RES_LVL_ST_TARG_ABOVE_WT*@SUM(day(I): WSF_RES_LVL_ST_TARG_ABOVE(I))+
WSF_RES_LVL_ST_TARG_BELOW_WT*@SUM(day(I): WSF_RES_LVL_ST_TARG_BELOW(I))+
WSF_RES_CMT_ST_TARG_WT*@SUM(day(I):  10*WSF_RES_CMT_ST_TARG_ABOVE(I) +
WSF_RES_CMT_ST_TARG_BELOW(I))+ !Added in the x10 to better represent ResSim model
behavior;
WSF_RES_BBK_ST_TARG_WT*@SUM(day(I): WSF_RES_BBK_ST_TARG_BELOW(I)) -
WSF_TOTAL_INCOME_WT * @SUM(day(I): WSF_TOTAL_INCOME(I)) +

!Farmington;
FAR_RES_OTI_ST_TARG_WT*@SUM(day(I): 100*FAR_RES_OTI_ST_TARG_ABOVE(I) +
FAR_RES_OTI_ST_TARG_BELOW(I)) + !100x weight is to better align with ResSim model;
FAR_RES_COL_ST_TARG_MAX_WT*@SUM(day(I): FAR_RES_COL_ST_TARG_MAX_ABOVE(I)) +
FAR_RES_WBR_ST_TARG_WT*@SUM(day(I): FAR_RES_WBR_ST_TARG_BELOW(I)) +
FAR_RES_LMD_ST_TARG_WT*@SUM(day(I): 100*FAR_RES_LMD_ST_TARG_ABOVE(I) +
FAR_RES_LMD_ST_TARG_BELOW(I)) + !100x weight is to better align with ResSim model;
FAR_RES_BKH_ST_TARG_WT*@SUM(day(I): FAR_RES_BKH_ST_TARG_ABOVE(I) +
FAR_RES_BKH_ST_TARG_BELOW(I)) +
FAR_RES_NEP_ST_TARG_WT*@SUM(day(I): FAR_RES_NEP_ST_TARG_ABOVE(I) +
FAR_RES_NEP_ST_TARG_BELOW(I)) +
FAR_RES_RBW_ST_TARG_WT*@SUM(day(I): FAR_RES_RBW_ST_TARG_ABOVE(I) +
FAR_RES_RBW_ST_TARG_BELOW(I)) +
FAR_RES_WBR_R_TARG_MIN_WT*@SUM(day(I): FAR_RES_WBR_R_TARG_MIN_BELOW(I)) +
FAR_RES_COL_R_TARG_MIN_WT*@SUM(day(I): FAR_RES_COL_R_TARG_MIN_BELOW(I)) +
FAR_RES_OTI_R_TARG_MAX_WT*@SUM(day(I): FAR_RES_OTI_R_TARG_MAX_ABOVE(I)) -
```

```
FAR_TOTAL_INCOME_WT * @SUM(day(I): FAR_TOTAL_INCOME(I)) +

!Lower Third Mainstem;
MAIN_RES_VRN_ST_TARG_WT * @SUM(day(I):  MAIN_RES_VRN_ST_TARG_ABOVE(I) +
MAIN_RES_VRN_ST_TARG_BELOW(I))+ !Vernon Storage Target;
MAIN_RES_TRN_ST_TARG_WT * @SUM(day(I):  MAIN_RES_TRN_ST_TARG_ABOVE(I) +
MAIN_RES_TRN_ST_TARG_BELOW(I))+ !Turners Falls Storage Target;
MAIN_RES_HOL_ST_TARG_WT * @SUM(day(I):  MAIN_RES_HOL_ST_TARG_ABOVE(I) +
MAIN_RES_HOL_ST_TARG_BELOW(I))- !Turners Falls Storage Target;
LOW3_TOTAL_INCOME_WT * @SUM(day(I): LOW3_TOTAL_INCOME(I)) +

!Upper;
UCN_RES_SCL_ST_END_WT*(UCN_RES_SCL_ST_END_ABOVE + UCN_RES_SCL_ST_END_BELOW) +
UCN_RES_FCL_ST_END_WT*(UCN_RES_FCL_ST_END_ABOVE + UCN_RES_FCL_ST_END_BELOW) +
UCN_RES_LFR_ST_END_WT*(UCN_RES_LFR_ST_END_ABOVE + UCN_RES_LFR_ST_END_BELOW) +
MAIN_RES_MOR_ST_END_WT*(MAIN_RES_MOR_ST_END_ABOVE + MAIN_RES_MOR_ST_END_BELOW) +
MAIN_RES_COM_ST_END_WT*(MAIN_RES_COM_ST_END_ABOVE + MAIN_RES_COM_ST_END_BELOW) +
MAIN_RES_MCD_ST_END_WT*(MAIN_RES_MCD_ST_END_ABOVE + MAIN_RES_MCD_ST_END_BELOW) +

!Middle;
OMP_RES_UNV_ST_END_WT*(OMP_RES_UNV_ST_END_ABOVE + OMP_RES_UNV_ST_END_BELOW) +
MAIN_RES_WLD_ST_END_WT*(MAIN_RES_WLD_ST_END_ABOVE + MAIN_RES_WLD_ST_END_BELOW) +
MAS_RES_GRF_ST_END_WT*(MAS_RES_GRF_ST_END_ABOVE + MAS_RES_GRF_ST_END_BELOW) +
MAS_RES_CRY_ST_END_WT*(MAS_RES_CRY_ST_END_ABOVE + MAS_RES_CRY_ST_END_BELOW) +
MAS_RES_GOO_ST_END_WT*(MAS_RES_GOO_ST_END_ABOVE + MAS_RES_GOO_ST_END_BELOW) +
MAS_RES_MSL_ST_END_WT*(MAS_RES_MSL_ST_END_ABOVE + MAS_RES_MSL_ST_END_BELOW) +
OTT_RES_NHD_ST_END_WT*(OTT_RES_NHD_ST_END_ABOVE + OTT_RES_NHD_ST_END_BELOW) +
SUG_RES_LSU_ST_END_WT*(SUG_RES_LSU_ST_END_ABOVE + SUG_RES_LSU_ST_END_BELOW) +
SUG_RES_SGR_ST_END_WT*(SUG_RES_SGR_ST_END_ABOVE + SUG_RES_SGR_ST_END_BELOW) +
BLK_RES_NSP_ST_END_WT*(BLK_RES_NSP_ST_END_ABOVE + BLK_RES_NSP_ST_END_BELOW) +
MAIN_RES_BFA_ST_END_WT*(MAIN_RES_BFA_ST_END_ABOVE + MAIN_RES_BFA_ST_END_BELOW) +

!West;
WST_RES_BMD_ST_END_WT*(WST_RES_BMD_ST_END_ABOVE + WST_RES_BMD_ST_END_BELOW) +
WST_RES_TWN_ST_END_WT*(WST_RES_TWN_ST_END_ABOVE + WST_RES_TWN_ST_END_BELOW) +

!Ashuelot;
ASH_RES_SMD_ST_END_WT*(ASH_RES_SMD_ST_END_ABOVE + ASH_RES_SMD_ST_END_BELOW) +
ASH_RES_OBD_ST_END_WT*(ASH_RES_OBD_ST_END_ABOVE + ASH_RES_OBD_ST_END_BELOW) +

!Millers;
MLR_RES_MON_ST_END_WT*(MLR_RES_MON_ST_END_ABOVE + MLR_RES_MON_ST_END_BELOW) +
MLR_RES_NEK_ST_END_WT*(MLR_RES_NEK_ST_END_ABOVE + MLR_RES_NEK_ST_END_BELOW) +
MLR_RES_BIR_ST_END_WT*(MLR_RES_BIR_ST_END_ABOVE + MLR_RES_BIR_ST_END_BELOW) +
MLR_RES_TUL_ST_END_WT*(MLR_RES_TUL_ST_END_ABOVE + MLR_RES_TUL_ST_END_BELOW) +

!Deefield;
DRF_RES_SOM_ST_END_WT*(DRF_RES_SOM_ST_END_ABOVE + DRF_RES_SOM_ST_END_BELOW) +
DRF_RES_SBG_ST_END_WT*(DRF_RES_SBG_ST_END_ABOVE + DRF_RES_SBG_ST_END_BELOW) +
DRF_RES_HAR_ST_END_WT*(DRF_RES_HAR_ST_END_ABOVE + DRF_RES_HAR_ST_END_BELOW) +
DRF_RES_SHR_ST_END_WT*(DRF_RES_SHR_ST_END_ABOVE + DRF_RES_SHR_ST_END_BELOW) +
DRF_RES_DV5_ST_END_WT*(DRF_RES_DV5_ST_END_ABOVE + DRF_RES_DV5_ST_END_BELOW) +
DRF_RES_FBR_ST_END_WT*(DRF_RES_FBR_ST_END_ABOVE + DRF_RES_FBR_ST_END_BELOW) +
DRF_RES_DV4_ST_END_WT*(DRF_RES_DV4_ST_END_ABOVE + DRF_RES_DV4_ST_END_BELOW) +
DRF_RES_DV3_ST_END_WT*(DRF_RES_DV3_ST_END_ABOVE + DRF_RES_DV3_ST_END_BELOW) +
DRF_RES_GRD_ST_END_WT*(DRF_RES_GRD_ST_END_ABOVE + DRF_RES_GRD_ST_END_BELOW) +
DRF_RES_DV2_ST_END_WT*(DRF_RES_DV2_ST_END_ABOVE + DRF_RES_DV2_ST_END_BELOW) +

!Chicopee;
CHP_RES_BFD_ST_END_WT*(CHP_RES_BFD_ST_END_ABOVE + CHP_RES_BFD_ST_END_BELOW) +
CHP_RES_CBD_ST_END_WT*(CHP_RES_CBD_ST_END_ABOVE + CHP_RES_CBD_ST_END_BELOW) +
CHP_RES_QWD_ST_END_WT*(CHP_RES_QWD_ST_END_ABOVE + CHP_RES_QWD_ST_END_BELOW) +
CHP_RES_MMBR_ST_END_WT*(CHP_RES_MMBR_ST_END_ABOVE + CHP_RES_MMBR_ST_END_BELOW) +
```

```
CHP_RES_RBD_ST_END_WT*(CHP_RES_RBD_ST_END_ABOVE + CHP_RES_RBD_ST_END_BELOW) +

!Westfield;
WSF_RES_KVL_ST_END_WT*(WSF_RES_KVL_ST_END_ABOVE + WSF_RES_KVL_ST_END_BELOW) +
WSF_RES_LVL_ST_END_WT*(WSF_RES_LVL_ST_END_ABOVE + WSF_RES_LVL_ST_END_BELOW) +
WSF_RES_CMT_ST_END_WT*(WSF_RES_CMT_ST_END_ABOVE + WSF_RES_CMT_ST_END_BELOW) +
WSF_RES_BBK_ST_END_WT*(WSF_RES_BBK_ST_END_ABOVE + WSF_RES_BBK_ST_END_BELOW) +

!Farmington;
FAR_RES_OTI_ST_END_WT*(FAR_RES_OTI_ST_END_ABOVE + FAR_RES_OTI_ST_END_BELOW) +
FAR_RES_COL_ST_END_WT*(FAR_RES_COL_ST_END_ABOVE + FAR_RES_COL_ST_END_BELOW) +
FAR_RES_WBR_ST_END_WT*(FAR_RES_WBR_ST_END_ABOVE + FAR_RES_WBR_ST_END_BELOW) +
FAR_RES_BKH_ST_END_WT*(FAR_RES_BKH_ST_END_ABOVE + FAR_RES_BKH_ST_END_BELOW) +
FAR_RES_LMD_ST_END_WT*(FAR_RES_LMD_ST_END_ABOVE + FAR_RES_LMD_ST_END_BELOW) +
FAR_RES_NEP_ST_END_WT*(FAR_RES_NEP_ST_END_ABOVE + FAR_RES_NEP_ST_END_BELOW) +
FAR_RES_RBW_ST_END_WT*(FAR_RES_RBW_ST_END_ABOVE + FAR_RES_RBW_ST_END_BELOW) +

!Lower;
MAIN_RES_VRN_ST_END_WT*(MAIN_RES_VRN_ST_END_ABOVE + MAIN_RES_VRN_ST_END_BELOW) +
MAIN_RES_TRN_ST_END_WT*(MAIN_RES_TRN_ST_END_ABOVE + MAIN_RES_TRN_ST_END_BELOW) +
MAIN_RES_HOL_ST_END_WT*(MAIN_RES_HOL_ST_END_ABOVE + MAIN_RES_HOL_ST_END_BELOW);
```

Constraints

Here constraints are defined which ensure that modeled hydropower operations mimic reality as closely as possible. Main constraints include a generalized mass balance which ensures that there is conservation of fluid mass over the modeled period and various reservoir operating constraints including physical reservoir storage limits and licensed minimum flow levels.

Mass Balance

```
!RESERVOIR MASS BALANCE;
```

Model Ramp Up

Since the Connecticut River is a large basin, it is broken up into discrete sections to provide a simplified routing scheme which limits how far water may travel in the period of a day. For the first few time periods, these *time-lags* are ignored so that the model may successfully 'ramp up' to operating capacity.

```
!Time lags are ignored in the first two time steps;
@FOR (day(I) | I #LE# 2:
!Upper Third;
[UCN_RES_SCL_MB1] UCN_RES_SCL_R(I) = UCN_RES_SCL_Q(I);
[UCN_RES_FCL_MB1] UCN_RES_FCL_R(I) = UCN_RES_FCL_Q(I)  + UCN_RES_SCL_R(I);
[UCN_RES_LFR_MB1] UCN_RES_LFR_R(I) = UCN_RES_LFR_Q(I)  + UCN_RES_FCL_R(I);
[MAIN_RES_MOR_MB1] MAIN_RES_MOR_R(I) = MAIN_RES_MOR_Q(I)  + UCN_RES_LFR_R(I);
[MAIN_RES_COM_MB1] MAIN_RES_COM_R(I) = MAIN_RES_COM_Q(I) + MAIN_RES_MOR_R(I);
[MAIN_RES_MCD_MB1] MAIN_RES_MCD_R(I) = MAIN_RES_MCD_Q(I) + MAIN_RES_COM_R(I);
!Middle Third;
[OMP_RES_UNV_MB1] OMP_RES_UNV_R(I) = OMP_RES_UNV_Q(I);
[MAIN_RES_WLD_MB1] MAIN_RES_WLD_R(I) = MAIN_RES_WLD_Q(I) + MAIN_RES_MCD_R(I) +
OMP_RES_UNV_R(I);
[MAS_RES_GOO_MB1] MAS_RES_GOO_R(I) = MAS_RES_GOO_Q(I);
[MAS_RES_CRY_MB1] MAS_RES_CRY_R(I) = MAS_RES_CRY_Q(I) + MAS_RES_GRF_R(I);
[MAS_RES_GRF_MB1] MAS_RES_GRF_R(I) = MAS_RES_GRF_Q(I);
[MAS_RES_MSL_MB1] MAS_RES_MSL_R(I) = MAS_RES_MSL_Q(I) + MAS_RES_CRY_R(I) +
MAS_RES_GOO_R(I);
[OTT_RES_NHD_MB1] OTT_RES_NHD_R(I) = OTT_RES_NHD_Q(I);
[SUG_RES_LSU_MB1] SUG_RES_LSU_R(I) = SUG_RES_LSU_Q(I);
```

```
[SUG_RES_SGR_MB1] SUG_RES_SGR_R(I) = SUG_RES_SGR_Q(I) + SUG_RES_LSU_R(I);
[BLK_RES_NSP_MB1] BLK_RES_NSP_R(I) = BLK_RES_NSP_Q(I);
[MAIN_RES_BFA_MB1] MAIN_RES_BFA_R(I) = MAIN_RES_BFA_Q(I) + MAIN_RES_WLD_R(I) +
MAS_RES_MSL_R(I) + OTT_RES_NHD_R(I) + SUG_RES_SGR_R(I)+ BLK_RES_NSP_R(I);
!West;
[WST_RES_BMD_MB1] WST_RES_BMD_R(I) = WST_RES_BMD_Q(I);
[WST_RES_TWN_MB1] WST_RES_TWN_R(I) = WST_RES_TWN_Q(I)  + WST_RES_BMD_R(I);
!Ashuelot;
[ASH_RES_SMD_MB1] ASH_RES_SMD_R(I) = ASH_RES_SMD_Q(I);
[ASH_RES_OBD_MB1] ASH_RES_OBD_R(I) = ASH_RES_OBD_Q(I);
!Millers;
[MLR_RES_MON_MB1] MLR_RES_MON_R(I) = MLR_RES_MON_Q(I);
[MLR_RES_NEK_MB1] MLR_RES_NEK_R(I) = MLR_RES_NEK_Q(I);
[MLR_RES_BIR_MB1] MLR_RES_BIR_R(I) = MLR_RES_BIR_Q(I) + MLR_RES_MON_R(I) +
MLR_RES_NEK_R(I);
[MLR_RES_TUL_MB1] MLR_RES_TUL_R(I) = MLR_RES_TUL_Q(I);
!Deerfield;
[DRF_RES_SOM_MB1] DRF_RES_SOM_R(I) = DRF_RES_SOM_Q(I);
[DRF_RES_SBG_MB1] DRF_RES_SBG_R(I) = DRF_RES_SBG_Q(I) + DRF_RES_SOM_R(I);
[DRF_RES_HAR_MB1] DRF_RES_HAR_R(I) = DRF_RES_HAR_Q(I) + DRF_RES_SBG_R(I);
[DRF_RES_SHR_MB1] DRF_RES_SHR_R(I) = DRF_RES_SHR_Q(I) + DRF_RES_HAR_R(I);
[DRF_RES_DV5_MB1] DRF_RES_DV5_R(I) = DRF_RES_DV5_Q(I) + DRF_RES_SHR_R(I);
[DRF_RES_FBR_MB1] DRF_RES_FBR_R(I) = DRF_RES_FBR_Q(I) + DRF_RES_DV5_R(I);
[DRF_RES_DV4_MB1] DRF_RES_DV4_R(I) = DRF_RES_DV4_Q(I) + DRF_RES_FBR_R(I);
[DRF_RES_DV3_MB1] DRF_RES_DV3_R(I) = DRF_RES_DV3_Q(I) + DRF_RES_DV4_R(I);
[DRF_RES_GRD_MB1] DRF_RES_GRD_R(I) = DRF_RES_GRD_Q(I) + DRF_RES_DV3_R(I);
[DRF_RES_DV2_MB1] DRF_RES_DV2_R(I) = DRF_RES_DV2_Q(I) + DRF_RES_GRD_R(I);
!Chicopee;
[CHP_RES_MMBR_MB1] CHP_RES_MMBR_R(I) = CHP_RES_MMBR_Q(I);
[CHP_RES_BFD_MB1] CHP_RES_BFD_R(I) =  CHP_RES_MMBR_R(I) + CHP_RES_BFD_Q(I);
[CHP_RES_CBD_MB1] CHP_RES_CBD_R(I) =  CHP_RES_CBD_Q(I);
[CHP_RES_QWD_MB1] CHP_RES_QWD_R(I) =  CHP_RES_QWD_Q(I);
[CHP_RES_RBD_MB1] CHP_RES_RBD_R(I) =  CHP_RES_RBD_Q(I) + CHP_RES_BFD_R(I) +
CHP_RES_CBD_R(I) + CHP_RES_QWD_R(I);
!Westfield;
[WSF_RES_KVL_MB1] WSF_RES_KVL_R(I) = WSF_RES_KVL_Q(I);
[WSF_RES_LVL_MB1] WSF_RES_LVL_R(I) = WSF_RES_LVL_Q(I);
[WSF_RES_BBK_MB1] WSF_RES_BBK_R(I) = WSF_RES_BBK_Q(I);
[WSF_RES_CMT_MB1] WSF_RES_CMT_R(I) = WSF_RES_BBK_R(I) + WSF_RES_CMT_Q(I);
!Farmington;
![FAR_RES_OTI_MB1] FAR_RES_OTI_R(I) = FAR_RES_OTI_Q(I);
![FAR_RES_COL_MB1] FAR_RES_COL_R(I) = FAR_RES_COL_Q(I) + FAR_RES_OTI_R(I);
![FAR_RES_WBR_MB1] FAR_RES_WBR_R(I) = FAR_RES_WBR_Q(I) + FAR_RES_COL_R(I);
![FAR_RES_BKH_MB1] FAR_RES_BKH_R(I) = FAR_RES_BKH_Q(I);
![FAR_RES_LMD_MB1] FAR_RES_LMD_R(I) = FAR_RES_LMD_Q(I) + FAR_RES_BKH_R(I) ;
![FAR_RES_NEP_MB1] FAR_RES_NEP_R(I) = FAR_RES_NEP_Q(I);
![FAR_RES_RBW_MB1] FAR_RES_RBW_R(I) = FAR_RES_RBW_Q(I) + FAR_RES_NEP_R(I) +
FAR_RES_LMD_R(I) + FAR_RES_WBR_R(I);
!Lower Third Mainstem;
[MAIN_RES_VRN_MB1] MAIN_RES_VRN_R(I) = MAIN_RES_VRN_Q(I) + WST_RES_TWN_R(I) +
MAIN_RES_BFA_R(I);
[MAIN_RES_TRN_MB1] MAIN_RES_TRN_R(I) = MAIN_RES_TRN_Q(I) + MAIN_RES_VRN_R(I) +
ASH_RES_SMD_R(I) + ASH_RES_OBD_R(I) + MLR_RES_TUL_R(I)+ MLR_RES_BIR_R(I);
[MAIN_RES_HOL_MB1] MAIN_RES_HOL_R(I) = MAIN_RES_HOL_Q(I) + MAIN_RES_TRN_R(I) +
DRF_RES_DV2_R(I);
);
```

Regular Mass Balance

Since the Connecticut River is a large basin, it is broken up into discrete sections to provide a simplified routing scheme which limits how far water may travel in the period of a day. After the

second modeled day of operation, these *time-lags* are used to ensure so that the model may successfully mimic real word routing.

```
!Greater than Day 2 - include the previous day's storage for each and also takes into
account time-lags in flow routing;
@FOR (day(I) | I #GE# 3:
!Upper Third;
[UCN_RES_SCL_MB2] UCN_RES_SCL_ST(I) = UCN_RES_SCL_ST(I-1) + UCN_RES_SCL_Q(I)  -
UCN_RES_SCL_R(I);
[UCN_RES_FCL_MB2] UCN_RES_FCL_ST(I) = UCN_RES_FCL_ST(I-1) + UCN_RES_FCL_Q(I)  +
UCN_RES_SCL_R(I) - UCN_RES_FCL_R(I);
[UCN_RES_LFR_MB2] UCN_RES_LFR_ST(I) = UCN_RES_LFR_ST(I-1) + UCN_RES_LFR_Q(I)  +
UCN_RES_FCL_R(I) - UCN_RES_LFR_R(I);
[MAIN_RES_MOR_MB2] MAIN_RES_MOR_ST(I) = MAIN_RES_MOR_ST(I-1) + MAIN_RES_MOR_Q(I)  +
UCN_RES_LFR_R(I-2) - MAIN_RES_MOR_R(I);    !2 day lag between Lake Francis and Moore;
[MAIN_RES_COM_MB2] MAIN_RES_COM_ST(I) = MAIN_RES_COM_ST(I-1) + MAIN_RES_COM_Q(I) +
MAIN_RES_MOR_R(I) - MAIN_RES_COM_R(I);
[MAIN_RES_MCD_MB2] MAIN_RES_MCD_ST(I) = MAIN_RES_MCD_ST(I-1) + MAIN_RES_MCD_Q(I) +
MAIN_RES_COM_R(I) - MAIN_RES_MCD_R(I);
!Middle Third;
[OMP_RES_UNV_MB2] OMP_RES_UNV_ST(I) = OMP_RES_UNV_ST(I-1) + OMP_RES_UNV_Q(I) -
OMP_RES_UNV_R(I);
[MAIN_RES_WLD_MB2] MAIN_RES_WLD_ST(I) = MAIN_RES_WLD_ST(I-1) + MAIN_RES_WLD_Q(I) +
MAIN_RES_MCD_R(I-1) + OMP_RES_UNV_R(I) - MAIN_RES_WLD_R(I);    !1 day lag between
Macindoes and Wilder;
[MAS_RES_GOO_MB2] MAS_RES_GOO_ST(I) = MAS_RES_GOO_ST(I-1) + MAS_RES_GOO_Q(I) -
MAS_RES_GOO_R(I);
[MAS_RES_CRY_MB2] MAS_RES_CRY_ST(I) = MAS_RES_CRY_ST(I-1) + MAS_RES_CRY_Q(I) +
MAS_RES_GRF_R(I) - MAS_RES_CRY_R(I);
[MAS_RES_GRF_MB2] MAS_RES_GRF_ST(I) = MAS_RES_GRF_ST(I-1) + MAS_RES_GRF_Q(I) -
MAS_RES_GRF_R(I);
[MAS_RES_MSL_MB2] MAS_RES_MSL_ST(I) = MAS_RES_MSL_ST(I-1) + MAS_RES_MSL_Q(I) +
MAS_RES_CRY_R(I) + MAS_RES_GOO_R(I) - MAS_RES_MSL_R(I);
[OTT_RES_NHD_MB2] OTT_RES_NHD_ST(I) = OTT_RES_NHD_ST(I-1) + OTT_RES_NHD_Q(I) -
OTT_RES_NHD_R(I);
[SUG_RES_LSU_MB2] SUG_RES_LSU_ST(I) = SUG_RES_LSU_ST(I-1) + SUG_RES_LSU_Q(I) -
SUG_RES_LSU_R(I);
[SUG_RES_SGR_MB2] SUG_RES_SGR_ST(I) = SUG_RES_SGR_ST(I-1) + SUG_RES_SGR_Q(I) +
SUG_RES_LSU_R(I) - SUG_RES_SGR_R(I);
[BLK_RES_NSP_MB2] BLK_RES_NSP_ST(I) = BLK_RES_NSP_ST(I-1) + BLK_RES_NSP_Q(I) -
BLK_RES_NSP_R(I);
[MAIN_RES_BFA_MB2] MAIN_RES_BFA_ST(I) = MAIN_RES_BFA_ST(I-1) + MAIN_RES_BFA_Q(I) +
MAIN_RES_WLD_R(I) + MAS_RES_MSL_R(I) + OTT_RES_NHD_R(I) + SUG_RES_SGR_R(I)+
BLK_RES_NSP_R(I) - MAIN_RES_BFA_R(I);
!West;
[WST_RES_BMD_MB2] WST_RES_BMD_ST (I) = WST_RES_BMD_ST(I-1) + WST_RES_BMD_Q(I) -
WST_RES_BMD_R(I);
[WST_RES_TWN_MB2] WST_RES_TWN_ST (I) = WST_RES_TWN_ST(I-1) + WST_RES_TWN_Q(I) +
WST_RES_BMD_R(I) - WST_RES_TWN_R(I);
!Ashuelot;
[ASH_RES_SMD_MB2] ASH_RES_SMD_ST(I) = ASH_RES_SMD_ST(I-1) + ASH_RES_SMD_Q(I) -
ASH_RES_SMD_R(I);
[ASH_RES_OBD_MB2] ASH_RES_OBD_ST(I) = ASH_RES_OBD_ST(I-1) + ASH_RES_OBD_Q(I) -
ASH_RES_OBD_R(I);
!Millers;
[MLR_RES_MON_MB2] MLR_RES_MON_ST(I) = MLR_RES_MON_ST(I-1) + MLR_RES_MON_Q(I) -
MLR_RES_MON_R(I);
[MLR_RES_NEK_MB2] MLR_RES_NEK_ST(I) = MLR_RES_NEK_ST(I-1) + MLR_RES_NEK_Q(I) -
MLR_RES_NEK_R(I);
[MLR_RES_BIR_MB2] MLR_RES_BIR_ST(I) = MLR_RES_BIR_ST(I-1) + MLR_RES_BIR_Q(I) -
MLR_RES_BIR_R(I) + MLR_RES_MON_R(I) + MLR_RES_NEK_R(I);
```

```
[MLR_RES_TUL_MB2] MLR_RES_TUL_ST(I) = MLR_RES_TUL_ST(I-1) + MLR_RES_TUL_Q(I) -
MLR_RES_TUL_R(I);
!Deerfield;
[DRF_RES_SOM_MB2] DRF_RES_SOM_ST(I) = DRF_RES_SOM_ST(I-1) + DRF_RES_SOM_Q(I) -
DRF_RES_SOM_R(I);
[DRF_RES_SBG_MB2] DRF_RES_SBG_ST(I) = DRF_RES_SBG_ST(I-1) + DRF_RES_SBG_Q(I) -
DRF_RES_SBG_R(I) + DRF_RES_SOM_R(I);
[DRF_RES_HAR_MB2] DRF_RES_HAR_ST(I) = DRF_RES_HAR_ST(I-1) + DRF_RES_HAR_Q(I) -
DRF_RES_HAR_R(I) + DRF_RES_SBG_R(I);
[DRF_RES_SHR_MB2] DRF_RES_SHR_ST(I) = DRF_RES_SHR_ST(I-1) + DRF_RES_SHR_Q(I) -
DRF_RES_SHR_R(I) + DRF_RES_HAR_R(I);
[DRF_RES_DV5_MB2] DRF_RES_DV5_ST(I) = DRF_RES_DV5_ST(I-1) + DRF_RES_DV5_Q(I) -
DRF_RES_DV5_R(I) + DRF_RES_SHR_R(I);
[DRF_RES_FBR_MB2] DRF_RES_FBR_ST(I) = DRF_RES_FBR_ST(I-1) + DRF_RES_FBR_Q(I) -
DRF_RES_FBR_R(I) + DRF_RES_DV5_R(I);
[DRF_RES_DV4_MB2] DRF_RES_DV4_ST(I) = DRF_RES_DV4_ST(I-1) + DRF_RES_DV4_Q(I) -
DRF_RES_DV4_R(I) + DRF_RES_FBR_R(I);
[DRF_RES_DV3_MB2] DRF_RES_DV3_ST(I) = DRF_RES_DV3_ST(I-1) + DRF_RES_DV3_Q(I) -
DRF_RES_DV3_R(I) + DRF_RES_DV4_R(I);
[DRF_RES_GRD_MB2] DRF_RES_GRD_ST(I) = DRF_RES_GRD_ST(I-1) + DRF_RES_GRD_Q(I) -
DRF_RES_GRD_R(I) + DRF_RES_DV3_R(I);
[DRF_RES_DV2_MB2] DRF_RES_DV2_ST(I) = DRF_RES_DV2_ST(I-1) + DRF_RES_DV2_Q(I) -
DRF_RES_DV2_R(I) + DRF_RES_GRD_R(I);
!Chicopee;
[CHP_RES_MMBR_MB2] CHP_RES_MMBR_ST(I) = CHP_RES_MMBR_ST(I-1) + CHP_RES_MMBR_Q(I) -
CHP_RES_MMBR_R(I) - CHP_RES_MMBR_W(I);
[CHP_RES_BFD_MB2] CHP_RES_BFD_ST(I) = CHP_RES_BFD_ST(I-1) + CHP_RES_MMBR_R(I) +
CHP_RES_BFD_Q(I) - CHP_RES_BFD_R(I);
[CHP_RES_CBD_MB2] CHP_RES_CBD_ST(I) =  CHP_RES_CBD_ST(I-1) + CHP_RES_CBD_Q(I) -
CHP_RES_CBD_R(I);
[CHP_RES_QWD_MB2] CHP_RES_QWD_ST(I) = CHP_RES_QWD_ST(I-1) + CHP_RES_QWD_Q(I) -
CHP_RES_QWD_R(I) - CHP_RES_QWD_W(I);
[CHP_RES_RBD_MB2] CHP_RES_RBD_ST(I) = CHP_RES_RBD_ST(I-1) + CHP_RES_RBD_Q(I) +
CHP_RES_BFD_R(I) + CHP_RES_CBD_R(I) + CHP_RES_QWD_R(I) - CHP_RES_RBD_R(I);
!Westfield;
[WSF_RES_KVL_MB2] WSF_RES_KVL_ST(I) = WSF_RES_KVL_ST(I-1) + WSF_RES_KVL_Q(I) -
WSF_RES_KVL_R(I);
[WSF_RES_LVL_MB2] WSF_RES_LVL_ST(I) = WSF_RES_LVL_ST(I-1) + WSF_RES_LVL_Q(I) -
WSF_RES_LVL_R(I);
[WSF_RES_BBK_MB2] WSF_RES_BBK_ST(I) = WSF_RES_BBK_ST(I-1) + WSF_RES_BBK_Q(I) -
WSF_RES_BBK_R(I);
[WSF_RES_CMT_MB2] WSF_RES_CMT_ST(I) = WSF_RES_CMT_ST(I-1) + WSF_RES_BBK_R(I) +
WSF_RES_CMT_Q(I) - WSF_RES_CMT_R(I) - WSF_RES_CMT_W(I);
!Farmington;
![FAR_RES_OTI_MB2] FAR_RES_OTI_ST(I) = FAR_RES_OTI_ST(I-1) + FAR_RES_OTI_Q(I) -
FAR_RES_OTI_R(I);
![FAR_RES_COL_MB2] FAR_RES_COL_ST(I) = FAR_RES_COL_ST(I-1) + FAR_RES_COL_Q(I) -
FAR_RES_COL_R(I) + FAR_RES_OTI_R(I);
![FAR_RES_WBR_MB2] FAR_RES_WBR_ST(I) = FAR_RES_WBR_ST(I-1) + FAR_RES_WBR_Q(I) -
FAR_RES_WBR_R(I) + FAR_RES_COL_R(I);
![FAR_RES_BKH_MB2] FAR_RES_BKH_ST(I) = FAR_RES_BKH_ST(I-1) + FAR_RES_BKH_Q(I) -
FAR_RES_BKH_R(I) - FAR_RES_BKH_W(I);
![FAR_RES_LMD_MB2] FAR_RES_LMD_ST(I) = FAR_RES_LMD_ST(I-1) + FAR_RES_LMD_Q(I) -
FAR_RES_LMD_R(I) + FAR_RES_BKH_R(I);
![FAR_RES_NEP_MB2] FAR_RES_NEP_ST(I) = FAR_RES_NEP_ST(I-1) + FAR_RES_NEP_Q(I) -
FAR_RES_NEP_R(I) - FAR_RES_NEP_W(I);
![FAR_RES_RBW_MB2] FAR_RES_RBW_ST(I) = FAR_RES_RBW_ST(I-1) + FAR_RES_RBW_Q(I) -
FAR_RES_RBW_R(I) + FAR_RES_NEP_R(I) + FAR_RES_LMD_R(I) + FAR_RES_WBR_R(I);
!Lower Third Mainstem;
[MAIN_RES_VRN_MB2] MAIN_RES_VRN_ST(I) = MAIN_RES_VRN_ST(I-1) + MAIN_RES_VRN_Q(I) +
WST_RES_TWN_R(I) + MAIN_RES_BFA_R(I-1) - MAIN_RES_VRN_R(I);       !1 day lag between
Bellows Falls and Vernon;
```

```
[MAIN_RES_TRN_MB2] MAIN_RES_TRN_ST(I) = MAIN_RES_TRN_ST(I-1) + MAIN_RES_TRN_Q(I) +
MAIN_RES_VRN_R(I) + ASH_RES_SMD_R(I) + ASH_RES_OBD_R(I) + MLR_RES_TUL_R(I)+
MLR_RES_BIR_R(I) - MAIN_RES_TRN_R(I);
[MAIN_RES_HOL_MB2] MAIN_RES_HOL_ST(I) = MAIN_RES_HOL_ST(I-1) + MAIN_RES_HOL_Q(I) +
MAIN_RES_TRN_R(I) + DRF_RES_DV2_R(I) - MAIN_RES_HOL_R(I);
);

@FOR (day(I) | I #GE# 2:
!Farmington;
[FAR_RES_OTI_MB2] FAR_RES_OTI_ST(I) = FAR_RES_OTI_ST(I-1) + FAR_RES_OTI_Q(I) -
FAR_RES_OTI_R(I);
[FAR_RES_COL_MB2] FAR_RES_COL_ST(I) = FAR_RES_COL_ST(I-1) + FAR_RES_COL_Q(I) -
FAR_RES_COL_R(I) + FAR_RES_OTI_R(I);
[FAR_RES_WBR_MB2] FAR_RES_WBR_ST(I) = FAR_RES_WBR_ST(I-1) + FAR_RES_WBR_Q(I) -
FAR_RES_WBR_R(I) + FAR_RES_COL_R(I);
[FAR_RES_BKH_MB2] FAR_RES_BKH_ST(I) = FAR_RES_BKH_ST(I-1) + FAR_RES_BKH_Q(I) -
FAR_RES_BKH_R(I) - FAR_RES_BKH_W(I);
[FAR_RES_LMD_MB2] FAR_RES_LMD_ST(I) = FAR_RES_LMD_ST(I-1) + FAR_RES_LMD_Q(I) -
FAR_RES_LMD_R(I) + FAR_RES_BKH_R(I);
[FAR_RES_NEP_MB2] FAR_RES_NEP_ST(I) = FAR_RES_NEP_ST(I-1) + FAR_RES_NEP_Q(I) -
FAR_RES_NEP_R(I) - FAR_RES_NEP_W(I);
[FAR_RES_RBW_MB2] FAR_RES_RBW_ST(I) = FAR_RES_RBW_ST(I-1) + FAR_RES_RBW_Q(I) -
FAR_RES_RBW_R(I) + FAR_RES_NEP_R(I) + FAR_RES_LMD_R(I) + FAR_RES_WBR_R(I);
);
```

Flood Checkpoint Mass Balance

```
!FLOOD CHECKPOINTS MASS BALANCE;

!Time lags are ignored in the first five time steps;
@FOR (day(I) | I #LE# 5:
[IND_G_IND_MB1] IND_G_IND(I) = IND_G_IND_Q(I) + UCN_RES_LFR_R(I);

[MAIN_G_WLEB_MB1] MAIN_G_WLEB(I) = MAIN_G_WLEB_Q(I)  +
        OMP_RES_UNV_R(I) +
        MAIN_RES_WLD_Q(I) +
                MAIN_RES_MCD_Q(I) + MAIN_RES_COM_Q(I) + MAIN_RES_MOR_Q(I) +
                UCN_RES_LFR_Q(I) + UCN_RES_FCL_Q(I) + UCN_RES_SCL_Q(I);

[MAIN_G_NWAL_MB1] MAIN_G_NWAL(I) = MAIN_G_NWAL_Q(I) +
        OMP_RES_UNV_R(I) + BLK_RES_NSP_R(I) + OTT_RES_NHD_R(I) +
                MAIN_RES_WLD_Q(I) + MAS_RES_GOO_Q(I) + MAS_RES_CRY_Q(I) +
MAS_RES_GRF_Q(I) + MAS_RES_MSL_Q(I) + SUG_RES_SGR_Q(I) + SUG_RES_LSU_Q(I) +
MAIN_RES_BFA_Q(I) +
        MAIN_RES_MCD_Q(I) + MAIN_RES_COM_Q(I) + MAIN_RES_MOR_Q(I) +
        UCN_RES_LFR_Q(I) + UCN_RES_FCL_Q(I) + UCN_RES_SCL_Q(I);

[MAIN_G_MONT_MB1] MAIN_G_MONT(I) = MAIN_G_MONT_Q(I) +
        WST_RES_TWN_R(I) + ASH_RES_SMD_R(I) + ASH_RES_OBD_R(I) + MLR_RES_TUL_R(I) +
MLR_RES_BIR_R(I) +
                OMP_RES_UNV_R(I) + BLK_RES_NSP_R(I) + OTT_RES_NHD_R(I) +
                MAIN_RES_VRN_Q(I) + MAIN_RES_TRN_Q(I) + DRF_RES_SOM_Q(I) +
DRF_RES_SBG_Q(I) +
        DRF_RES_HAR_Q(I) + DRF_RES_SHR_Q(I) + DRF_RES_DV5_Q(I) + DRF_RES_FBR_Q(I) +
DRF_RES_DV4_Q(I) + DRF_RES_DV3_Q(I) + DRF_RES_GRD_Q(I) + DRF_RES_DV2_Q(I) +
                MAIN_RES_WLD_Q(I) + MAS_RES_GOO_Q(I) + MAS_RES_CRY_Q(I) +
MAS_RES_GRF_Q(I) + MAS_RES_MSL_Q(I) + SUG_RES_SGR_Q(I) + SUG_RES_LSU_Q(I) +
MAIN_RES_BFA_Q(I) +
        MAIN_RES_MCD_Q(I) + MAIN_RES_COM_Q(I) + MAIN_RES_MOR_Q(I) +
        UCN_RES_LFR_Q(I) + UCN_RES_FCL_Q(I) + UCN_RES_SCL_Q(I);

[ASH_G_HIN_MB1] ASH_G_HIN(I) = ASH_G_HIN_Q(I) +
```

```
      ASH_RES_SMD_R(I) + ASH_RES_OBD_R(I);

[MLR_G_ERV_MB1] MLR_G_ERV(I) = MLR_G_ERV_Q(I) +
      MLR_RES_TUL_R(I) + MLR_RES_BIR_R(I);

[CHP_G_GIB_MB1] CHP_G_GIB(I) = CHP_G_GIB_Q(I) +
                CHP_RES_BFD_R(I);

[CHP_G_IND_MB1] CHP_G_IND(I) = CHP_G_IND_Q(I) +
                CHP_RES_BFD_R(I) + CHP_RES_CBD_R(I) +
      CHP_RES_QWD_Q(I) + CHP_RES_RBD_Q(I);

[WSF_G_WSF_MB1] WSF_G_WSF(I) = WSF_G_WSF_Q(I) +
      WSF_RES_KVL_R(I) + WSF_RES_LVL_R(I) +
      WSF_RES_CMT_Q(I) + WSF_RES_BBK_Q(I);

[FAR_G_RIV_MB1] FAR_G_RIV(I) = FAR_G_RIV_Q(I) +
      FAR_RES_COL_R(I) +
      FAR_RES_WBR_Q(I);

[FAR_G_UNV_MB1] FAR_G_UNV(I) = FAR_G_UNV_Q(I) +
      FAR_RES_COL_R(I) +
      FAR_RES_WBR_Q(I) + FAR_RES_BKH_Q(I) + FAR_RES_LMD_Q(I) + FAR_RES_NEP_Q(I) -
FAR_RES_BKH_W(I) - FAR_RES_NEP_W(I);

[MAIN_G_THOM_MB1] MAIN_G_THOM(I) = MAIN_G_THOM_Q(I) +
      WSF_RES_KVL_R(I) + WSF_RES_LVL_R(I) + CHP_RES_BFD_R(I) + CHP_RES_CBD_R(I) +
      WST_RES_TWN_R(I) + ASH_RES_SMD_R(I) + ASH_RES_OBD_R(I) + MLR_RES_TUL_R(I) +
MLR_RES_BIR_R(I) +
                OMP_RES_UNV_R(I) + BLK_RES_NSP_R(I) + OTT_RES_NHD_R(I) +
   CHP_G_IND_Q(I) + WSF_G_WSF_Q(I) +
   WSF_RES_BBK_Q(I) + WSF_RES_CMT_Q(I) - WSF_RES_CMT_W(I) + CHP_RES_QWD_Q(I) -
CHP_RES_QWD_W(I) +  CHP_RES_RBD_Q(I) +
                MAIN_RES_VRN_Q(I) + MAIN_RES_TRN_Q(I) + DRF_RES_SOM_Q(I) +
DRF_RES_SBG_Q(I) +
      DRF_RES_HAR_Q(I) + DRF_RES_SHR_Q(I) + DRF_RES_DV5_Q(I) + DRF_RES_FBR_Q(I) +
DRF_RES_DV4_Q(I) + DRF_RES_DV3_Q(I) + DRF_RES_GRD_Q(I) + DRF_RES_DV2_Q(I) +
                MAIN_RES_WLD_Q(I) + MAS_RES_GOO_Q(I) + MAS_RES_CRY_Q(I) +
MAS_RES_GRF_Q(I) + MAS_RES_MSL_Q(I) + SUG_RES_SGR_Q(I) + SUG_RES_LSU_Q(I) +
MAIN_RES_BFA_Q(I) +
      MAIN_RES_MCD_Q(I) + MAIN_RES_COM_Q(I) + MAIN_RES_MOR_Q(I) +
      UCN_RES_LFR_Q(I) + UCN_RES_FCL_Q(I) + UCN_RES_SCL_Q(I);

[MAIN_G_HART_MB1] MAIN_G_HART(I) = MAIN_G_HART_Q(I) +
      FAR_RES_COL_R(I) +
      WSF_RES_KVL_R(I) + WSF_RES_LVL_R(I) + CHP_RES_BFD_R(I) + CHP_RES_CBD_R(I) +
      WST_RES_TWN_R(I) + ASH_RES_SMD_R(I) + ASH_RES_OBD_R(I) + MLR_RES_TUL_R(I) +
MLR_RES_BIR_R(I) +
                OMP_RES_UNV_R(I) + BLK_RES_NSP_R(I) + OTT_RES_NHD_R(I) +
   MAIN_G_THOM_Q(I) +
      FAR_RES_WBR_Q(I) + FAR_RES_BKH_Q(I) + FAR_RES_LMD_Q(I) + FAR_RES_NEP_Q(I) +
FAR_RES_RBW_Q(I) +  MAIN_RES_HOL_Q(I)  - FAR_RES_BKH_W(I) - FAR_RES_NEP_W(I) +
      WSF_RES_BBK_Q(I) + WSF_RES_CMT_Q(I) - WSF_RES_CMT_W(I) + CHP_RES_QWD_Q(I) -
CHP_RES_QWD_W(I) +  CHP_RES_RBD_Q(I) +
                MAIN_RES_VRN_Q(I) + MAIN_RES_TRN_Q(I) + DRF_RES_SOM_Q(I) +
DRF_RES_SBG_Q(I) +
      DRF_RES_HAR_Q(I) + DRF_RES_SHR_Q(I) + DRF_RES_DV5_Q(I) + DRF_RES_FBR_Q(I) +
DRF_RES_DV4_Q(I) + DRF_RES_DV3_Q(I) + DRF_RES_GRD_Q(I) + DRF_RES_DV2_Q(I) +
                MAIN_RES_WLD_Q(I) + MAS_RES_GOO_Q(I) + MAS_RES_CRY_Q(I) +
MAS_RES_GRF_Q(I) + MAS_RES_MSL_Q(I) + SUG_RES_SGR_Q(I) + SUG_RES_LSU_Q(I) +
MAIN_RES_BFA_Q(I) +
      MAIN_RES_MCD_Q(I) + MAIN_RES_COM_Q(I) + MAIN_RES_MOR_Q(I) +
      UCN_RES_LFR_Q(I) + UCN_RES_FCL_Q(I) + UCN_RES_SCL_Q(I);
```

```
);

@FOR (day(I) | I #GE# 6:

[IND_G_IND_MB2] IND_G_IND(I) = IND_G_IND_Q(I) + UCN_RES_LFR_R(I);

[MAIN_G_WLEB_MB2] MAIN_G_WLEB(I) = MAIN_G_WLEB_Q(I)  +
        OMP_RES_UNV_R(I) +
                  MAIN_RES_WLD_Q(I) +
                  MAIN_RES_MCD_Q(I-1) + MAIN_RES_COM_Q(I-1) + MAIN_RES_MOR_Q(I-1) +
                  UCN_RES_LFR_Q(I-3) + UCN_RES_FCL_Q(I-3) + UCN_RES_SCL_Q(I-3);

[MAIN_G_NWAL_MB2] MAIN_G_NWAL(I) = MAIN_G_NWAL_Q(I) +
                  OMP_RES_UNV_R(I) + BLK_RES_NSP_R(I) + OTT_RES_NHD_R(I) +
                  MAIN_RES_WLD_Q(I) + MAS_RES_GOO_Q(I) + MAS_RES_CRY_Q(I) +
MAS_RES_GRF_Q(I) + MAS_RES_MSL_Q(I) + SUG_RES_SGR_Q(I) + SUG_RES_LSU_Q(I) +
MAIN_RES_BFA_Q(I) +
        MAIN_RES_MCD_Q(I-1) + MAIN_RES_COM_Q(I-1) + MAIN_RES_MOR_Q(I-1) +
        UCN_RES_LFR_Q(I-3) + UCN_RES_FCL_Q(I-3) + UCN_RES_SCL_Q(I-3);

[MAIN_G_MONT_MB2] MAIN_G_MONT(I) = MAIN_G_MONT_Q(I) +
        WST_RES_TWN_R(I) + ASH_RES_SMD_R(I) + ASH_RES_OBD_R(I) + MLR_RES_TUL_R(I) +
MLR_RES_BIR_R(I) +
                  OMP_RES_UNV_R(I-1) + BLK_RES_NSP_R(I-1) + OTT_RES_NHD_R(I-1) +
                  MAIN_RES_VRN_Q(I) + MAIN_RES_TRN_Q(I) + DRF_RES_SOM_Q(I) +
DRF_RES_SBG_Q(I) +
        DRF_RES_HAR_Q(I) + DRF_RES_SHR_Q(I) + DRF_RES_DV5_Q(I) + DRF_RES_FBR_Q(I) +
DRF_RES_DV4_Q(I) + DRF_RES_DV3_Q(I) + DRF_RES_GRD_Q(I) + DRF_RES_DV2_Q(I) +
                  MAIN_RES_WLD_Q(I-1) + MAS_RES_GOO_Q(I-1) + MAS_RES_CRY_Q(I-1) +
MAS_RES_GRF_Q(I-1) + MAS_RES_MSL_Q(I-1) + SUG_RES_SGR_Q(I-1) + SUG_RES_LSU_Q(I-1) +
MAIN_RES_BFA_Q(I-1) +
        MAIN_RES_MCD_Q(I-2) + MAIN_RES_COM_Q(I-2) + MAIN_RES_MOR_Q(I-2) +
        UCN_RES_LFR_Q(I-4) + UCN_RES_FCL_Q(I-4) + UCN_RES_SCL_Q(I-4);

[ASH_G_HIN_MB2] ASH_G_HIN(I) = ASH_G_HIN_Q(I) +
      ASH_RES_SMD_R(I) + ASH_RES_OBD_R(I);

[MLR_G_ERV_MB2] MLR_G_ERV(I) = MLR_G_ERV_Q(I) +
      MLR_RES_TUL_R(I) + MLR_RES_BIR_R(I);

[CHP_G_GIB_MB2] CHP_G_GIB(I) = CHP_G_GIB_Q(I) +
                CHP_RES_BFD_R(I);

[CHP_G_IND_MB2] CHP_G_IND(I) = CHP_G_IND_Q(I) +
                  CHP_RES_BFD_R(I) + CHP_RES_CBD_R(I) +
           CHP_RES_QWD_Q(I) + CHP_RES_RBD_Q(I);

[WSF_G_WSF_MB2] WSF_G_WSF(I) = WSF_G_WSF_Q(I) +
      WSF_RES_KVL_R(I) + WSF_RES_LVL_R(I) +
      WSF_RES_CMT_Q(I) + WSF_RES_BBK_Q(I);

[FAR_G_RIV_MB2] FAR_G_RIV(I) = FAR_G_RIV_Q(I) +
      FAR_RES_COL_R(I) +
      FAR_RES_WBR_Q(I);

[FAR_G_UNV_MB2] FAR_G_UNV(I) = FAR_G_UNV_Q(I) +
      FAR_RES_COL_R(I) +
      FAR_RES_WBR_Q(I) + FAR_RES_BKH_Q(I) + FAR_RES_LMD_Q(I) + FAR_RES_NEP_Q(I) -
FAR_RES_BKH_W(I) - FAR_RES_NEP_W(I);

[MAIN_G_THOM_MB2] MAIN_G_THOM(I) = MAIN_G_THOM_Q(I) +
        WSF_RES_KVL_R(I) + WSF_RES_LVL_R(I) + CHP_RES_BFD_R(I) + CHP_RES_CBD_R(I) +
```

```
        WST_RES_TWN_R(I-1) + ASH_RES_SMD_R(I-1) + ASH_RES_OBD_R(I-1) +
MLR_RES_TUL_R(I-1) + MLR_RES_BIR_R(I-1) +
                  OMP_RES_UNV_R(I-2) + BLK_RES_NSP_R(I-2) + OTT_RES_NHD_R(I-2) +
   CHP_G_IND_Q(I) + WSF_G_WSF_Q(I) +
        WSF_RES_BBK_Q(I) + WSF_RES_CMT_Q(I) - WSF_RES_CMT_W(I) + CHP_RES_QWD_Q(I) -
CHP_RES_QWD_W(I) +  CHP_RES_RBD_Q(I) +
                  MAIN_RES_VRN_Q(I-1) + MAIN_RES_TRN_Q(I-1) + DRF_RES_SOM_Q(I-1) +
DRF_RES_SBG_Q(I-1) +
        DRF_RES_HAR_Q(I-1) + DRF_RES_SHR_Q(I-1) + DRF_RES_DV5_Q(I-1) +
DRF_RES_FBR_Q(I-1) + DRF_RES_DV4_Q(I-1) + DRF_RES_DV3_Q(I-1) + DRF_RES_GRD_Q(I-1) +
DRF_RES_DV2_Q(I-1) +
                  MAIN_RES_WLD_Q(I-2) + MAS_RES_GOO_Q(I-2) + MAS_RES_CRY_Q(I-2) +
MAS_RES_GRF_Q(I-2) + MAS_RES_MSL_Q(I-2) + SUG_RES_SGR_Q(I-2) + SUG_RES_LSU_Q(I-2) +
MAIN_RES_BFA_Q(I-2) +
             MAIN_RES_MCD_Q(I-3) + MAIN_RES_COM_Q(I-3) + MAIN_RES_MOR_Q(I-3) +
        UCN_RES_LFR_Q(I-5) + UCN_RES_FCL_Q(I-5) + UCN_RES_SCL_Q(I-5);

[MAIN_G_HART_MB2] MAIN_G_HART(I) = MAIN_G_HART_Q(I) +
        FAR_RES_COL_R(I) +
        WSF_RES_KVL_R(I) + WSF_RES_LVL_R(I) + CHP_RES_BFD_R(I) + CHP_RES_CBD_R(I) +
        WST_RES_TWN_R(I) + ASH_RES_SMD_R(I) + ASH_RES_OBD_R(I) + MLR_RES_TUL_R(I) +
MLR_RES_BIR_R(I) +
                  OMP_RES_UNV_R(I) + BLK_RES_NSP_R(I) + OTT_RES_NHD_R(I) +
   MAIN_G_THOM_Q(I) +
        FAR_RES_WBR_Q(I) + FAR_RES_BKH_Q(I) + FAR_RES_LMD_Q(I) + FAR_RES_NEP_Q(I) +
FAR_RES_RBW_Q(I) +  MAIN_RES_HOL_Q(I)  - FAR_RES_BKH_W(I) - FAR_RES_NEP_W(I) +
        WSF_RES_BBK_Q(I) + WSF_RES_CMT_Q(I) - WSF_RES_CMT_W(I) + CHP_RES_QWD_Q(I) -
CHP_RES_QWD_W(I) +  CHP_RES_RBD_Q(I) +
                  MAIN_RES_VRN_Q(I-1) + MAIN_RES_TRN_Q(I-1) + DRF_RES_SOM_Q(I-1) +
DRF_RES_SBG_Q(I-1) +
        DRF_RES_HAR_Q(I-1) + DRF_RES_SHR_Q(I-1) + DRF_RES_DV5_Q(I-1) +
DRF_RES_FBR_Q(I-1) + DRF_RES_DV4_Q(I-1) + DRF_RES_DV3_Q(I-1) + DRF_RES_GRD_Q(I-1) +
DRF_RES_DV2_Q(I-1) +
                  MAIN_RES_WLD_Q(I-2) + MAS_RES_GOO_Q(I-2) + MAS_RES_CRY_Q(I-2) +
MAS_RES_GRF_Q(I-2) + MAS_RES_MSL_Q(I-2) + SUG_RES_SGR_Q(I-2) + SUG_RES_LSU_Q(I-2) +
MAIN_RES_BFA_Q(I-2) +
        MAIN_RES_MCD_Q(I-3) + MAIN_RES_COM_Q(I-3) + MAIN_RES_MOR_Q(I-3) +
        UCN_RES_LFR_Q(I-5) + UCN_RES_FCL_Q(I-5) + UCN_RES_SCL_Q(I-5);
);


!USGS CHECKPOINTS MASS BALANCE;
@FOR (day(I):
[IND_USGS_IND_MOD_MB] IND_USGS_IND_MOD(I) = IND_G_IND_Q(I) + UCN_RES_LFR_R(I);

[MAIN_USGS_DAL_MOD_MB] MAIN_USGS_DAL_MOD(I) = MAIN_RES_MOR_Q(I) + UCN_RES_LFR_R(I);

[MAIN_USGS_WEL_MOD_MB] MAIN_USGS_WEL_MOD(I) = MAIN_RES_MCD_R(I);

[MAIN_USGS_WLEB_MOD_MB] MAIN_USGS_WLEB_MOD(I) = MAIN_G_WLEB_Q(I) + MAIN_RES_WLD_R(I);

[MAS_USGS_MAS_MOD_MB] MAS_USGS_MAS_MOD(I) = MAS_RES_MSL_R(I);

[SUG_USGS_CLR_MOD_MB] SUG_USGS_CLR_MOD(I) = ((269-45)/45)*SUG_RES_LSU_Q(I) +
SUG_RES_LSU_R(I);

[MAIN_USGS_NWAL_MOD_MB] MAIN_USGS_NWAL_MOD(I) = MAIN_G_NWAL_Q(I) + MAIN_RES_BFA_R(I);

[MAIN_USGS_MONT_MOD_MB] MAIN_USGS_MONT_MOD(I) = MAIN_G_MONT_Q(I) + MAIN_RES_TRN_R(I) +
DRF_RES_DV2_R(I);

[DRF_USGS_CRM_MOD_MB] DRF_USGS_CRM_MOD(I) = DRF_RES_FBR_R(I) + .5*DRF_RES_DV4_Q(I);
```

```
[DRF_USGS_DRF_MOD_MB] DRF_USGS_DRF_MOD(I) = DRF_RES_DV2_R(I);

[MAIN_USGS_HOL_MOD_MB] MAIN_USGS_HOL_MOD(I) = MAIN_RES_HOL_R(I);

[CHP_USGS_IND_MOD_MB] CHP_USGS_IND_MOD(I) = CHP_G_IND_Q(I) + CHP_RES_RBD_R(I);

[CHP_USGS_SWF_MOD_MB] CHP_USGS_SWF_MOD(I) = CHP_RES_QWD_R(I);

[WSF_USGS_WSF_MOD_MB] WSF_USGS_WSF_MOD(I) = WSF_G_WSF_Q(I) + WSF_RES_KVL_R(I) +
WSF_RES_LVL_R(I) + WSF_RES_CMT_R(I);

![FAR_USGS_OTI_MOD_MB] FAR_USGS_OTI_MOD(I) = FAR_RES_OTI_R(I);

![FAR_USGS_UNV_MOD_MB] FAR_USGS_UNV_MOD(I) =  FAR_G_UNV_Q(I) + FAR_RES_WBR_R(I) +
FAR_RES_LMD_R(I) + FAR_RES_NEP_R(I);

![FAR_USGS_RBW_MOD_MB] FAR_USGS_RBW_MOD(I) = FAR_RES_RBW_R(I);

[MAIN_USGS_THOM_MOD_MB] MAIN_USGS_THOM_MOD(I) = MAIN_G_THOM_Q(I) + WSF_G_WSF_Q(I) +
CHP_G_IND_Q(I) + MAIN_RES_HOL_R(I) + WSF_RES_KVL_R(I) + WSF_RES_LVL_R(I) +
WSF_RES_CMT_R(I) + CHP_RES_RBD_R(I);

[CHP_USGS_GIBBS_MOD_MB] CHP_USGS_GIBBS_MOD(I) = CHP_G_GIB_Q(I) + CHP_RES_BFD_R(I);

[MLR_USGS_ERV_MOD_MB] MLR_USGS_ERV_MOD(I) =  MLR_G_ERV_Q(I) + MLR_RES_TUL_R(I) +
MLR_RES_BIR_R(I);

[ASH_USGS_HIN_MOD_MB] ASH_USGS_HIN_MOD(I) = ASH_G_HIN_Q(I) + ASH_RES_SMD_R(I) +
ASH_RES_OBD_R(I);

![FAR_USGS_RIV_MOD_MB] FAR_USGS_RIV_MOD(I) = FAR_G_RIV_Q(I) + FAR_RES_WBR_R(I);

);

!---CONTSTRAINTS AND TARGETS---;
```

Storage Limits

Here the physical capacities of each dam are defined as volumetric boundaries ranging from maximum operational/allowed storage to empty.

```
!Storage Limits;
!Upper Third;
@FOR(day(I):
[UNC_RES_SCL_ST_BOUND] @BND(0,UCN_RES_SCL_ST,UCN_RES_SCL_OPP_RANGE_UPPER);
[UNC_RES_FCL_ST_BOUND]
@BND(UCN_RES_FCL_OPP_RANGE_LOWER,UCN_RES_FCL_ST,UCN_RES_FCL_OPP_RANGE_UPPER);
[UNC_RES_LFR_ST_BOUND]
@BND(UCN_RES_LFR_OPP_RANGE_LOWER,UCN_RES_LFR_ST,UCN_RES_LFR_OPP_RANGE_UPPER);
[MAIN_RES_MOR_ST_BOUND]
@BND(MAIN_RES_MOR_OPP_RANGE_LOWER,MAIN_RES_MOR_ST,MAIN_RES_MOR_ST_MAX);
[MAIN_RES_COM_ST_BOUND]
@BND(MAIN_RES_COM_OPP_RANGE_LOWER,MAIN_RES_COM_ST,MAIN_RES_COM_ST_MAX);
[MAIN_RES_MCD_ST_BOUND] @BND(0,MAIN_RES_MCD_ST,MAIN_RES_MCD_ST_MAX);
!Middle Third;
[BLK_RES_NSP_ST_BOUND] @BND(0,BLK_RES_NSP_ST,BLK_RES_NSP_ST_MAX);
[MAS_RES_GOO_ST_BOUND] @BND(0,MAS_RES_GOO_ST,MAS_RES_GOO_ST_MAX);
[MAS_RES_CRY_ST_BOUND] @BND(0,MAS_RES_CRY_ST,MAS_RES_CRY_ST_MAX);
[MAS_RES_GRF_ST_BOUND] @BND(0,MAS_RES_GRF_ST,MAS_RES_GRF_ST_MAX);
[MAS_RES_MSL_ST_BOUND] @BND(0,MAS_RES_MSL_ST,MAS_RES_MSL_ST_MAX);
[OMP_RES_UNV_ST_BOUND] @BND(0,OMP_RES_UNV_ST,OMP_RES_UNV_ST_MAX);
[OTT_RES_NHD_ST_BOUND] @BND(0,OTT_RES_NHD_ST,OTT_RES_NHD_ST_MAX);
```

```
[SUG_RES_LSU_ST_BOUND] @BND(0,SUG_RES_LSU_ST,SUG_RES_LSU_ST_MAX);
[SUG_RES_SGR_ST_BOUND] @BND(0,SUG_RES_SGR_ST,SUG_RES_SGR_ST_MAX);
[MAIN_RES_BFA_ST_BOUND] @BND(0,MAIN_RES_BFA_ST,MAIN_RES_BFA_ST_MAX);
[MAIN_RES_WLD_ST_BOUND] @BND(0,MAIN_RES_WLD_ST,MAIN_RES_WLD_ST_MAX);
!West;
[WST_RES_BMD_ST_BOUND] @BND(0,WST_RES_BMD_ST,WST_RES_BMD_ST_MAX);
[WST_RES_TWN_ST_BOUND] @BND(0,WST_RES_TWN_ST,WST_RES_TWN_ST_MAX);
!Ashuelot;
[ASH_RES_SMD_ST_BOUND] @BND(0,ASH_RES_SMD_ST,ASH_RES_SMD_ST_MAX);
[ASH_RES_OBD_ST_BOUND] @BND(0,ASH_RES_OBD_ST,ASH_RES_OBD_ST_MAX);
!Millers;
[MLR_RES_MON_ST_BOUND] @BND(0,MLR_RES_MON_ST,MLR_RES_MON_ST_MAX);
[MLR_RES_NEK_ST_BOUND] @BND(0,MLR_RES_NEK_ST,MLR_RES_NEK_ST_MAX);
[MLR_RES_BIR_ST_BOUND] @BND(0,MLR_RES_BIR_ST,MLR_RES_BIR_ST_MAX);
[MLR_RES_TUL_ST_BOUND] @BND(0,MLR_RES_TUL_ST,MLR_RES_TUL_ST_MAX);
!Deerfield;
[DRF_RES_SOM_ST_BOUND]
@BND(DRF_RES_SOM_OPP_RANGE_LOWER,DRF_RES_SOM_ST,DRF_RES_SOM_ST_MAX);
[DRF_RES_SBG_ST_BOUND] @BND(0,DRF_RES_SBG_ST,DRF_RES_SBG_ST_MAX);
[DRF_RES_HAR_ST_BOUND]
@BND(DRF_RES_HAR_OPP_RANGE_LOWER,DRF_RES_HAR_ST,DRF_RES_HAR_ST_MAX);
[DRF_RES_SHR_ST_BOUND] @BND(0,DRF_RES_SHR_ST,DRF_RES_SHR_ST_MAX);
[DRF_RES_FBR_ST_BOUND] @BND(0,DRF_RES_FBR_ST,DRF_RES_FBR_ST_MAX);
[DRF_RES_DV5_ST_BOUND] @BND(0,DRF_RES_DV5_ST,DRF_RES_DV5_ST_MAX);
[DRF_RES_DV4_ST_BOUND] @BND(0,DRF_RES_DV4_ST,DRF_RES_DV4_ST_MAX);
[DRF_RES_DV3_ST_BOUND] @BND(0,DRF_RES_DV3_ST,DRF_RES_DV3_ST_MAX);
[DRF_RES_GRD_ST_BOUND] @BND(0,DRF_RES_GRD_ST,DRF_RES_GRD_ST_MAX);
[DRF_RES_DV2_ST_BOUND] @BND(0,DRF_RES_DV2_ST,DRF_RES_DV2_ST_MAX);
!Chicopee;
[CHP_RES_BFD_ST_BOUND] @BND(0,CHP_RES_BFD_ST,CHP_RES_BFD_ST_MAX);
[CHP_RES_CBD_ST_BOUND] @BND(0,CHP_RES_CBD_ST,CHP_RES_CBD_ST_MAX);
[CHP_RES_QWD_ST_BOUND] @BND(0,CHP_RES_QWD_ST,CHP_RES_QWD_ST_MAX);
[CHP_RES_MMBR_ST_BOUND] @BND(0,CHP_RES_MMBR_ST,CHP_RES_MMBR_ST_MAX);
[CHP_RES_RBD_ST_BOUND] @BND(0,CHP_RES_RBD_ST,CHP_RES_RBD_ST_MAX);
!Westfield;
[WSF_RES_KVL_ST_BOUND] @BND(0,WSF_RES_KVL_ST,WSF_RES_KVL_ST_MAX);
[WSF_RES_LVL_ST_BOUND] @BND(WSF_RES_LVL_ST_MIN,WSF_RES_LVL_ST,WSF_RES_LVL_ST_MAX);
[WSF_RES_CMT_ST_BOUND] @BND(WSF_RES_CMT_ST_MIN,WSF_RES_CMT_ST,WSF_RES_CMT_ST_MAX);
[WSF_RES_BBK_ST_BOUND] @BND(0,WSF_RES_BBK_ST,WSF_RES_BBK_ST_MAX);
!Farmington;
[FAR_RES_OTI_ST_BOUND] @BND(FAR_RES_OTI_ST_MIN,FAR_RES_OTI_ST,FAR_RES_OTI_ST_MAX);
[FAR_RES_COL_ST_BOUND] @BND(0,FAR_RES_COL_ST,FAR_RES_COL_ST_MAX); !The 1,000 af
sedimentation pool is not included in the max st level--add it in post processing if
necessary;
[FAR_RES_WBR_ST_BOUND] @BND(0,FAR_RES_WBR_ST,FAR_RES_WBR_ST_MAX);
[FAR_RES_NEP_ST_BOUND] @BND(FAR_RES_NEP_ST_MIN,FAR_RES_NEP_ST,FAR_RES_NEP_ST_MAX);
[FAR_RES_LMD_ST_BOUND] @BND(0,FAR_RES_LMD_ST,FAR_RES_LMD_ST_MAX);
[FAR_RES_BKH_ST_BOUND] @BND(FAR_RES_BKH_ST_MIN,FAR_RES_BKH_ST,FAR_RES_BKH_ST_MAX);
[FAR_RES_RBW_ST_BOUND] @BND(FAR_RES_RBW_ST_MIN,FAR_RES_RBW_ST,FAR_RES_RBW_ST_MAX);
!Lower Thirs Mainstem;
[MAIN_RES_VRN_ST_BOUND] @BND(0,MAIN_RES_VRN_ST,MAIN_RES_VRN_ST_MAX);
[MAIN_RES_TRN_ST_BOUND] @BND(0,MAIN_RES_TRN_ST,MAIN_RES_TRN_ST_MAX);
[MAIN_RES_HOL_ST_BOUND] @BND(0,MAIN_RES_HOL_ST,MAIN_RES_HOL_ST_MAX);
);

!Seasonal Storage Limits;
@FOR(day(I)|I #GE# 141 #AND# I #LE# 365:
!Upper Third;
[MAIN_RES_MOR_SUMMER_BOUND]
@BND(MAIN_RES_MOR_SUMMER_OPP_RANGE_LOWER,MAIN_RES_MOR_ST(I),MAIN_RES_MOR_ST_MAX);
[MAIN_RES_COM_SUMMER_BOUND]
@BND(MAIN_RES_COM_SUMMER_OPP_RANGE_LOWER,MAIN_RES_COM_ST(I),MAIN_RES_COM_ST_MAX);
```

```
!Deerfield;
[DRF_RES_SOM_SUMMER_BOUND]
@BND(DRF_RES_SOM_SUMMER_OPP_RANGE_LOWER,DRF_RES_SOM_ST(I),DRF_RES_SOM_ST_MAX);
[DRF_RES_HAR_SUMMER_BOUND]
@BND(DRF_RES_HAR_SUMMER_OPP_RANGE_LOWER,DRF_RES_HAR_ST(I),DRF_RES_HAR_ST_MAX);

);
```

Variable Definitions

The following lines of code are mathematical definitions which compare modeled and expected flows in order to inform the objective function. For instance, the `[IND_FLOOD_IND_TARG]` line defines the difference between modeled flow and the warning flood flow target at the Indian River gage such that positive values are stored in ABOVE term and the negatives are stored in the BELOW term.

```
!Targets to Optimize;
@FOR (day(I) | I #GE# 1:
!Flood Gauge Target 1;
[IND_FLOOD_IND_TARG]  IND_G_IND(I) - IND_G_IND_TARG = IND_G_IND_TARG_ABOVE(I) -
IND_G_IND_TARG_BELOW(I);    !Indian Gauge at Indian Gauge;
[MAIN_FLOOD_WLEB_TARG]  MAIN_G_WLEB(I) - MAIN_G_WLEB_TARG =  MAIN_G_WLEB_TARG_ABOVE(I)
- MAIN_G_WLEB_TARG_BELOW(I);   !West Lebanon Gauge;
[MAIN_FLOOD_NWAL_TARG]  MAIN_G_NWAL(I) - MAIN_G_NWAL_TARG =  MAIN_G_NWAL_TARG_ABOVE(I)
- MAIN_G_NWAL_TARG_BELOW(I);   !North Walpole Gauge;
[MAIN_FLOOD_MONT_TARG]  MAIN_G_MONT(I) - MAIN_G_MONT_TARG =  MAIN_G_MONT_TARG_ABOVE(I)
- MAIN_G_MONT_TARG_BELOW(I);   !Montague City Gauge;
[ASH_FLOOD_HIN_TARG] ASH_G_HIN(I) - ASH_G_HIN_TARG = ASH_G_HIN_TARG_ABOVE(I) -
ASH_G_HIN_TARG_BELOW(I);     !Hinsdale Gauge;
[MLR_FLOOD_ERV_TARG] MLR_G_ERV(I) - MLR_G_ERV_TARG = MLR_G_ERV_TARG_ABOVE(I) -
MLR_G_ERV_TARG_BELOW(I);     !Erving Gauge;
[CHP_FLOOD_GIB_TARG] CHP_G_GIB(I) - CHP_G_GIB_TARG = CHP_G_GIB_TARG_ABOVE(I) -
CHP_G_GIB_TARG_BELOW(I);     !Gibbs Crossing Gauge;
[CHP_FLOOD_IND_TARG] CHP_G_IND(I) - CHP_G_IND_TARG = CHP_G_IND_TARG_ABOVE(I) -
CHP_G_IND_TARG_BELOW(I);     !Indian Gauge in Chicopee Basin;
[WSF_FLOOD_WSF_TARG] WSF_G_WSF(I) - WSF_G_WSF_TARG = WSF_G_WSF_TARG_ABOVE(I) -
WSF_G_WSF_TARG_BELOW(I);     !Westfield Gauge;
[FAR_FLOOD_RIV_TARG]  FAR_G_RIV(I) - FAR_G_RIV_TARG = FAR_G_RIV_TARG_ABOVE(I) -
FAR_G_RIV_TARG_BELOW(I);    !Riverton in Farmington;
[FAR_FLOOD_UNV_TARG]  FAR_G_UNV(I) - FAR_G_UNV_TARG = FAR_G_UNV_TARG_ABOVE(I) -
FAR_G_UNV_TARG_BELOW(I);    !Unionville in Farmington;
[MAIN_FLOOD_THOM_TARG] MAIN_G_THOM(I) - MAIN_G_THOM_TARG =  MAIN_G_THOM_TARG_ABOVE(I)
- MAIN_G_THOM_TARG_BELOW(I);   !Thompsonville;
[MAIN_FLOOD_HART_TARG]  MAIN_G_HART(I) - MAIN_G_HART_TARG =  MAIN_G_HART_TARG_ABOVE(I)
- MAIN_G_HART_TARG_BELOW(I);   !Hartford;

!Flood Gage Target 2;
[IND_FLOOD_IND_TARG2]  IND_G_IND(I) - IND_G_IND_TARG2 = IND_G_IND_TARG2_ABOVE(I) -
IND_G_IND_TARG2_BELOW(I);    !Indian Gauge at Indian Gauge;
[MAIN_FLOOD_WLEB_TARG2]  MAIN_G_WLEB(I) - MAIN_G_WLEB_TARG2 =
MAIN_G_WLEB_TARG2_ABOVE(I) - MAIN_G_WLEB_TARG2_BELOW(I);   !West Lebanon Gauge;
[MAIN_FLOOD_NWAL_TARG2]  MAIN_G_NWAL(I) - MAIN_G_NWAL_TARG2 =
MAIN_G_NWAL_TARG2_ABOVE(I) - MAIN_G_NWAL_TARG2_BELOW(I);   !North Walpole Gauge;
[MAIN_FLOOD_MONT_TARG2]  MAIN_G_MONT(I) - MAIN_G_MONT_TARG2 =
MAIN_G_MONT_TARG2_ABOVE(I) - MAIN_G_MONT_TARG2_BELOW(I);   !Montague City Gauge;
[ASH_FLOOD_HIN_TARG2] ASH_G_HIN(I) - ASH_G_HIN_TARG2 = ASH_G_HIN_TARG2_ABOVE(I) -
ASH_G_HIN_TARG2_BELOW(I);     !Hinsdale Gauge;
[MLR_FLOOD_ERV_TARG2] MLR_G_ERV(I) - MLR_G_ERV_TARG2 = MLR_G_ERV_TARG2_ABOVE(I) -
MLR_G_ERV_TARG2_BELOW(I);     !Erving Gauge;
```

```
[CHP_FLOOD_GIB_TARG2] CHP_G_GIB(I) - CHP_G_GIB_TARG2 = CHP_G_GIB_TARG2_ABOVE(I) -
CHP_G_GIB_TARG2_BELOW(I);     !Gibbs Crossing Gauge;
[CHP_FLOOD_IND_TARG2] CHP_G_IND(I) - CHP_G_IND_TARG2 = CHP_G_IND_TARG2_ABOVE(I) -
CHP_G_IND_TARG2_BELOW(I);     !Indian Gauge in Chicopee Basin;
[WSF_FLOOD_WSF_TARG2] WSF_G_WSF(I) - WSF_G_WSF_TARG2 = WSF_G_WSF_TARG2_ABOVE(I) -
WSF_G_WSF_TARG2_BELOW(I);     !Westfield Gauge;
[FAR_FLOOD_RIV_TARG2]  FAR_G_RIV(I) - FAR_G_RIV_TARG2 = FAR_G_RIV_TARG2_ABOVE(I) -
FAR_G_RIV_TARG2_BELOW(I);    !Riverton in Farmington;
[FAR_FLOOD_UNV_TARG2]  FAR_G_UNV(I) - FAR_G_UNV_TARG2 = FAR_G_UNV_TARG2_ABOVE(I) -
FAR_G_UNV_TARG2_BELOW(I);    !Unionville in Farmington;
[MAIN_FLOOD_THOM_TARG2] MAIN_G_THOM(I) - MAIN_G_THOM_TARG2 =
MAIN_G_THOM_TARG2_ABOVE(I) - MAIN_G_THOM_TARG2_BELOW(I);   !Thompsonville;
[MAIN_FLOOD_HART_TARG2]  MAIN_G_HART(I) - MAIN_G_HART_TARG2 =
MAIN_G_HART_TARG2_ABOVE(I) - MAIN_G_HART_TARG2_BELOW(I);   !Hartford;

!Flood USACE Storage Targets;
[OMP_RES_UNV_ST_FLOOD_TARG1] OMP_RES_UNV_ST(I) - PERCENT*OMP_RES_UNV_ST_MAX =
OMP_RES_UNV_ST_FLOOD_TARG_ABOVE(I) - OMP_RES_UNV_ST_FLOOD_TARG_BELOW(I);
[OTT_RES_NHD_ST_FLOOD_TARG1] OTT_RES_NHD_ST(I) - PERCENT*OTT_RES_NHD_ST_MAX =
OTT_RES_NHD_ST_FLOOD_TARG_ABOVE(I) - OTT_RES_NHD_ST_FLOOD_TARG_BELOW(I);
[BLK_RES_NSP_ST_FLOOD_TARG1] BLK_RES_NSP_ST(I) - PERCENT*BLK_RES_NSP_ST_MAX =
BLK_RES_NSP_ST_FLOOD_TARG_ABOVE(I) - BLK_RES_NSP_ST_FLOOD_TARG_BELOW(I);
[WST_RES_BMD_ST_FLOOD_TARG1] WST_RES_BMD_ST(I) - PERCENT*WST_RES_BMD_ST_MAX =
WST_RES_BMD_ST_FLOOD_TARG_ABOVE(I) - WST_RES_BMD_ST_FLOOD_TARG_BELOW(I);
[WST_RES_TWN_ST_FLOOD_TARG1] WST_RES_TWN_ST(I) - PERCENT*WST_RES_TWN_ST_MAX =
WST_RES_TWN_ST_FLOOD_TARG_ABOVE(I) - WST_RES_TWN_ST_FLOOD_TARG_BELOW(I);
[ASH_RES_SMD_ST_FLOOD_TARG1] ASH_RES_SMD_ST(I) - PERCENT*ASH_RES_SMD_ST_MAX =
ASH_RES_SMD_ST_FLOOD_TARG_ABOVE(I) - ASH_RES_SMD_ST_FLOOD_TARG_BELOW(I);
[ASH_RES_OBD_ST_FLOOD_TARG1] ASH_RES_OBD_ST(I) - PERCENT*ASH_RES_OBD_ST_MAX =
ASH_RES_OBD_ST_FLOOD_TARG_ABOVE(I) - ASH_RES_OBD_ST_FLOOD_TARG_BELOW(I);
[MLR_RES_TUL_ST_FLOOD_TARG1] MLR_RES_TUL_ST(I) - PERCENT*MLR_RES_TUL_ST_MAX =
MLR_RES_TUL_ST_FLOOD_TARG_ABOVE(I) - MLR_RES_TUL_ST_FLOOD_TARG_BELOW(I);
[MLR_RES_BIR_ST_FLOOD_TARG1] MLR_RES_BIR_ST(I) - PERCENT*MLR_RES_BIR_ST_MAX =
MLR_RES_BIR_ST_FLOOD_TARG_ABOVE(I) - MLR_RES_BIR_ST_FLOOD_TARG_BELOW(I);
[WSF_RES_KVL_ST_FLOOD_TARG1] WSF_RES_KVL_ST(I) - PERCENT*WSF_RES_KVL_ST_MAX =
WSF_RES_KVL_ST_FLOOD_TARG_ABOVE(I) - WSF_RES_KVL_ST_FLOOD_TARG_BELOW(I);
[WSF_RES_LVL_ST_FLOOD_TARG1] WSF_RES_LVL_ST(I) - (PERCENT*WSF_RES_LVL_ST_MAX +
9400*43560*(1-PERCENT)) = WSF_RES_LVL_ST_FLOOD_TARG_ABOVE(I) -
WSF_RES_LVL_ST_FLOOD_TARG_BELOW(I);   !We account for the water supply pool to isolate
storage above PERCENT% of flood control storage;
[CHP_RES_BFD_ST_FLOOD_TARG1] CHP_RES_BFD_ST(I) - PERCENT*CHP_RES_BFD_ST_MAX =
CHP_RES_BFD_ST_FLOOD_TARG_ABOVE(I) - CHP_RES_BFD_ST_FLOOD_TARG_BELOW(I);
[CHP_RES_CBD_ST_FLOOD_TARG1] CHP_RES_CBD_ST(I) - PERCENT*CHP_RES_CBD_ST_MAX =
CHP_RES_CBD_ST_FLOOD_TARG_ABOVE(I) - CHP_RES_CBD_ST_FLOOD_TARG_BELOW(I);
![FAR_RES_COL_ST_FLOOD_TARG1] FAR_RES_COL_ST(I) - (PERCENT*(50242*43560) +
(5000+30691)*43560) = FAR_RES_COL_ST_FLOOD_TARG_ABOVE(I) -
FAR_RES_COL_ST_FLOOD_TARG_BELOW(I);
);

!Storage Targets;
!Upper Third;
@FOR (day(I) | I #GE# 1:
[UCN_RES_SCL_ST_TARG_L]     UCN_RES_SCL_ST(I) - UCN_RES_SCL_ST_TARG(I) =
UCN_RES_SCL_ST_TARG_ABOVE(I) - UCN_RES_SCL_ST_TARG_BELOW(I); !First CT Lake Target
Storage;
[UCN_RES_FCL_ST_TARG_L]     UCN_RES_FCL_ST(I) - UCN_RES_FCL_ST_TARG(I) =
UCN_RES_FCL_ST_TARG_ABOVE(I) - UCN_RES_FCL_ST_TARG_BELOW(I);  !Second CT Lake Target
Storage;
[UCN_RES_LFR_ST_TARG_L]     UCN_RES_LFR_ST(I) - UCN_RES_LFR_ST_TARG(I) =
UCN_RES_LFR_ST_TARG_ABOVE(I) - UCN_RES_LFR_ST_TARG_BELOW(I);  !Lake Francis Target
Storage;
```

```
[MAIN_RES_MOR_ST_TARG_L]    MAIN_RES_MOR_ST(I) - MAIN_RES_MOR_ST_TARG(I) =
MAIN_RES_MOR_ST_TARG_ABOVE(I) - MAIN_RES_MOR_ST_TARG_BELOW(I); !Moore's Target
Storage;
[MAIN_RES_COM_ST_TARG_L]    MAIN_RES_COM_ST(I) - MAIN_RES_COM_ST_TARG(I) =
MAIN_RES_COM_ST_TARG_ABOVE(I) - MAIN_RES_COM_ST_TARG_BELOW(I); !Moore's Target
Storage;
[MAIN_RES_MCD_ST_TARG_L]    MAIN_RES_MCD_ST(I) - MAIN_RES_MCD_ST_TARG(I) =
MAIN_RES_MCD_ST_TARG_ABOVE(I) - MAIN_RES_MCD_ST_TARG_BELOW(I); !Moore's Target
Storage;

!Middle Third;
[BLK_RES_NSP_ST_TARG_L]     BLK_RES_NSP_ST(I) - BLK_RES_NSP_ST_TARG(I) =
BLK_RES_NSP_ST_TARG_ABOVE(I) - BLK_RES_NSP_ST_TARG_BELOW(I);
[MAS_RES_GOO_ST_TARG_L]     MAS_RES_GOO_ST(I) - MAS_RES_GOO_ST_TARG(I) =
MAS_RES_GOO_ST_TARG_ABOVE(I) - MAS_RES_GOO_ST_TARG_BELOW(I);
[MAS_RES_CRY_ST_TARG_L]     MAS_RES_CRY_ST(I) - MAS_RES_CRY_ST_TARG(I) =
MAS_RES_CRY_ST_TARG_ABOVE(I) - MAS_RES_CRY_ST_TARG_BELOW(I);
[MAS_RES_GRF_ST_TARG_L]     MAS_RES_GRF_ST(I) - MAS_RES_GRF_ST_TARG(I) =
MAS_RES_GRF_ST_TARG_ABOVE(I) - MAS_RES_GRF_ST_TARG_BELOW(I);
[MAS_RES_MSL_ST_TARG_L]     MAS_RES_MSL_ST(I) - MAS_RES_MSL_ST_TARG(I) =
MAS_RES_MSL_ST_TARG_ABOVE(I) - MAS_RES_MSL_ST_TARG_BELOW(I);
[OMP_RES_UNV_ST_TARG_L]     OMP_RES_UNV_ST(I) - OMP_RES_UNV_ST_TARG(I) =
OMP_RES_UNV_ST_TARG_ABOVE(I) - OMP_RES_UNV_ST_TARG_BELOW(I);
[OTT_RES_NHD_ST_TARG_L]     OTT_RES_NHD_ST(I) - OTT_RES_NHD_ST_TARG(I) =
OTT_RES_NHD_ST_TARG_ABOVE(I) - OTT_RES_NHD_ST_TARG_BELOW(I);
[SUG_RES_LSU_ST_TARG_L]     SUG_RES_LSU_ST(I) - SUG_RES_LSU_ST_TARG(I) =
SUG_RES_LSU_ST_TARG_ABOVE(I) - SUG_RES_LSU_ST_TARG_BELOW(I);
[SUG_RES_SGR_ST_TARG_L]     SUG_RES_SGR_ST(I) - SUG_RES_SGR_ST_TARG(I) =
SUG_RES_SGR_ST_TARG_ABOVE(I) - SUG_RES_SGR_ST_TARG_BELOW(I);
[MAIN_RES_BFA_ST_TARG_L]    MAIN_RES_BFA_ST(I) - MAIN_RES_BFA_ST_TARG(I) =
MAIN_RES_BFA_ST_TARG_ABOVE(I) - MAIN_RES_BFA_ST_TARG_BELOW(I);
[MAIN_RES_WLD_ST_TARG_L]    MAIN_RES_WLD_ST(I) - MAIN_RES_WLD_ST_TARG(I) =
MAIN_RES_WLD_ST_TARG_ABOVE(I) - MAIN_RES_WLD_ST_TARG_BELOW(I);


!West;
[WST_RES_BMD_ST_TARG_L]  WST_RES_BMD_ST(I) - WST_RES_BMD_ST_TARG(I) =
WST_RES_BMD_ST_TARG_ABOVE(I) - WST_RES_BMD_ST_TARG_BELOW(I);
[WST_RES_TWN_ST_TARG_L]  WST_RES_TWN_ST(I) - WST_RES_TWN_ST_TARG(I) =
WST_RES_TWN_ST_TARG_ABOVE(I) - WST_RES_TWN_ST_TARG_BELOW(I);

!Ashuelot;
[ASH_RES_SMD_ST_TARG_L] ASH_RES_SMD_ST(I) - ASH_RES_SMD_ST_TARG(I) =
ASH_RES_SMD_ST_TARG_ABOVE(I) - ASH_RES_SMD_ST_TARG_BELOW(I);
[ASH_RES_OBD_ST_TARG_L] ASH_RES_OBD_ST(I) - ASH_RES_OBD_ST_TARG(I) =
ASH_RES_OBD_ST_TARG_ABOVE(I) - ASH_RES_OBD_ST_TARG_BELOW(I);

!Millers;
[MLR_RES_MON_ST_TARG_L] MLR_RES_MON_ST(I)  - MLR_RES_MON_ST_TARG(I) =
MLR_RES_MON_ST_TARG_ABOVE(I) - MLR_RES_MON_ST_TARG_BELOW(I);
[MLR_RES_NEK_ST_TARG_L] MLR_RES_NEK_ST(I)  - MLR_RES_NEK_ST_TARG(I) =
MLR_RES_NEK_ST_TARG_ABOVE(I) - MLR_RES_NEK_ST_TARG_BELOW(I);
[MLR_RES_BIR_ST_TARG_L] MLR_RES_BIR_ST(I)  - MLR_RES_BIR_ST_TARG(I) =
MLR_RES_BIR_ST_TARG_ABOVE(I) - MLR_RES_BIR_ST_TARG_BELOW(I);
[MLR_RES_TUL_ST_TARG_L] MLR_RES_TUL_ST(I) - MLR_RES_TUL_ST_TARG(I) =
MLR_RES_TUL_ST_TARG_ABOVE(I) - MLR_RES_TUL_ST_TARG_BELOW(I);

!Deerfield;
[DRF_RES_SOM_ST_TARG_L]     DRF_RES_SOM_ST(I) - DRF_RES_SOM_ST_TARG(I) =
DRF_RES_SOM_ST_TARG_ABOVE(I) - DRF_RES_SOM_ST_TARG_BELOW(I);
[DRF_RES_HAR_ST_TARG_L]     DRF_RES_HAR_ST(I) - DRF_RES_HAR_ST_TARG(I) =
DRF_RES_HAR_ST_TARG_ABOVE(I) - DRF_RES_HAR_ST_TARG_BELOW(I);
```

```
[DRF_RES_SBG_ST_TARG_L]     DRF_RES_SBG_ST(I) - DRF_RES_SBG_ST_TARG(I) =
DRF_RES_SBG_ST_TARG_ABOVE(I) - DRF_RES_SBG_ST_TARG_BELOW(I);
[DRF_RES_SHR_ST_TARG_L]     DRF_RES_SHR_ST(I) - DRF_RES_SHR_ST_TARG(I) =
DRF_RES_SHR_ST_TARG_ABOVE(I) - DRF_RES_SHR_ST_TARG_BELOW(I);
[DRF_RES_FBR_ST_TARG_L]     DRF_RES_FBR_ST(I) - DRF_RES_FBR_ST_TARG(I) =
DRF_RES_FBR_ST_TARG_ABOVE(I) - DRF_RES_FBR_ST_TARG_BELOW(I);
[DRF_RES_DV5_ST_TARG_L]     DRF_RES_DV5_ST(I) - DRF_RES_DV5_ST_TARG(I) =
DRF_RES_DV5_ST_TARG_ABOVE(I) - DRF_RES_DV5_ST_TARG_BELOW(I);
[DRF_RES_DV4_ST_TARG_L]     DRF_RES_DV4_ST(I) - DRF_RES_DV4_ST_TARG(I) =
DRF_RES_DV4_ST_TARG_ABOVE(I) - DRF_RES_DV4_ST_TARG_BELOW(I);
[DRF_RES_DV3_ST_TARG_L]     DRF_RES_DV3_ST(I) - DRF_RES_DV3_ST_TARG(I) =
DRF_RES_DV3_ST_TARG_ABOVE(I) - DRF_RES_DV3_ST_TARG_BELOW(I);
[DRF_RES_GRD_ST_TARG_L]     DRF_RES_GRD_ST(I) - DRF_RES_GRD_ST_TARG(I) =
DRF_RES_GRD_ST_TARG_ABOVE(I) - DRF_RES_GRD_ST_TARG_BELOW(I);
[DRF_RES_DV2_ST_TARG_L]     DRF_RES_DV2_ST(I) - DRF_RES_DV2_ST_TARG(I) =
DRF_RES_DV2_ST_TARG_ABOVE(I) - DRF_RES_DV2_ST_TARG_BELOW(I);

!Chicopee;
[CHP_RES_BFD_ST_TARG_L]  CHP_RES_BFD_ST(I) - CHP_RES_BFD_ST_TARG(I) =
CHP_RES_BFD_ST_TARG_ABOVE(I) - CHP_RES_BFD_ST_TARG_BELOW(I);    !Barre Falls Storage
Target;
[CHP_RES_CBD_ST_TARG_L]  CHP_RES_CBD_ST(I) - CHP_RES_CBD_ST_TARG(I) =
CHP_RES_CBD_ST_TARG_ABOVE(I) - CHP_RES_CBD_ST_TARG_BELOW(I);    !Conant Brook Storage
Target;
[CHP_RES_QWD_ST_TARG_L]  CHP_RES_QWD_ST(I) - CHP_RES_QWD_ST_TARG(I) =
CHP_RES_QWD_ST_TARG_ABOVE(I) - CHP_RES_QWD_ST_TARG_BELOW(I);    !Quabbin Storage
Target;
[CHP_RES_MMBR_ST_TARG_L] CHP_RES_MMBR_ST(I) - CHP_RES_MMBR_ST_TARG(I) =
CHP_RES_MMBR_ST_TARG_ABOVE(I) - CHP_RES_MMBR_ST_TARG_BELOW(I);   !Mare Meadow/Bickford
Storage Target;
[CHP_RES_RBD_ST_TARG_L]  CHP_RES_RBD_ST(I) - CHP_RES_RBD_ST_TARG(I) =
CHP_RES_RBD_ST_TARG_ABOVE(I) - CHP_RES_RBD_ST_TARG_BELOW(I);

!Westfield;
[WSF_RES_KVL_ST_TARG_L]    WSF_RES_KVL_ST(I) - WSF_RES_KVL_ST_TARG(I) =
WSF_RES_KVL_ST_TARG_ABOVE(I) - WSF_RES_KVL_ST_TARG_BELOW(I);
[WSF_RES_LVL_ST_TARG_L]    WSF_RES_LVL_ST(I) - WSF_RES_LVL_ST_TARG(I) =
WSF_RES_LVL_ST_TARG_ABOVE(I) - WSF_RES_LVL_ST_TARG_BELOW(I);
[WSF_RES_CMT_ST_TARG_L]    WSF_RES_CMT_ST(I) - WSF_RES_CMT_ST_TARG(I) =
WSF_RES_CMT_ST_TARG_ABOVE(I) - WSF_RES_CMT_ST_TARG_BELOW(I);
[WSF_RES_BBK_ST_TARG_L]    WSF_RES_BBK_ST(I) - WSF_RES_BBK_ST_TARG(I) =
WSF_RES_BBK_ST_TARG_ABOVE(I) - WSF_RES_BBK_ST_TARG_BELOW(I);

!Farmington;
[FAR_RES_OTI_ST_TARG_L]   FAR_RES_OTI_ST(I) - FAR_RES_OTI_ST_TARG(I) =
FAR_RES_OTI_ST_TARG_ABOVE(I) - FAR_RES_OTI_ST_TARG_BELOW(I);
[FAR_RES_COL_ST_TARG_MAX_L] FAR_RES_COL_ST(I) - FAR_RES_COL_ST_TARG_MAX(I) =
FAR_RES_COL_ST_TARG_MAX_ABOVE(I) - FAR_RES_COL_ST_TARG_MAX_BELOW(I);   !We minimize
the above target variable to ensure spring shad DEP pool doesn't fill until after Apr
21 and empties by Jul 1;
[FAR_RES_WBR_ST_TARG_L]   FAR_RES_WBR_ST(I) - FAR_RES_WBR_ST_TARG(I) =
FAR_RES_WBR_ST_TARG_ABOVE(I) - FAR_RES_WBR_ST_TARG_BELOW(I);
[FAR_RES_LMD_ST_TARG_L]   FAR_RES_LMD_ST(I) - FAR_RES_LMD_ST_TARG(I) =
FAR_RES_LMD_ST_TARG_ABOVE(I) - FAR_RES_LMD_ST_TARG_BELOW(I);
[FAR_RES_BKH_ST_TARG_L]   FAR_RES_BKH_ST(I) - FAR_RES_BKH_ST_TARG(I) =
FAR_RES_BKH_ST_TARG_ABOVE(I) - FAR_RES_BKH_ST_TARG_BELOW(I);
[FAR_RES_NEP_ST_TARG_L]   FAR_RES_NEP_ST(I) - FAR_RES_NEP_ST_TARG(I) =
FAR_RES_NEP_ST_TARG_ABOVE(I) - FAR_RES_NEP_ST_TARG_BELOW(I);
[FAR_RES_RBW_ST_TARG_L]   FAR_RES_RBW_ST(I) - FAR_RES_RBW_ST_TARG(I) =
FAR_RES_RBW_ST_TARG_ABOVE(I) - FAR_RES_RBW_ST_TARG_BELOW(I);

!Lower Third Mainstem;
```

```
[MAIN_RES_VRN_ST_TARG_L] MAIN_RES_VRN_ST(I) - MAIN_RES_VRN_ST_TARG(I) =
MAIN_RES_VRN_ST_TARG_ABOVE(I) - MAIN_RES_VRN_ST_TARG_BELOW(I);
[MAIN_RES_TRN_ST_TARG_L] MAIN_RES_TRN_ST(I) - MAIN_RES_TRN_ST_TARG(I) =
MAIN_RES_TRN_ST_TARG_ABOVE(I) - MAIN_RES_TRN_ST_TARG_BELOW(I);
[MAIN_RES_HOL_ST_TARG_L] MAIN_RES_HOL_ST(I) - MAIN_RES_HOL_ST_TARG(I) =
MAIN_RES_HOL_ST_TARG_ABOVE(I) - MAIN_RES_HOL_ST_TARG_BELOW(I);
);
```

Ecological/Minimum Flow Constraints

This section defines ecological flow and storage constraints which force reservoirs to operate according to mandated seasonal or year-long operating rules.

```
!Ecological Constraints Related to Storage;
!Upper Third;
[MAIN_RES_MOR_ST_TARG_MIN_BASS_L] MAIN_RES_MOR_ST(141) > 200175*43560;    !Day 141 is
May 21;
[MAIN_RES_COM_ST_TARG_MIN_BASS_L] MAIN_RES_COM_ST(141) > 27016*43560;    !Day 141 is
May 21;
@FOR(day(I)| I #GE# 142 #AND# I #LE# 181:                                  !Day 142 is
May 22, Day 181 is June 30;
[MAIN_RES_MOR_ST_TARG_DD_BASS_L1] MAIN_RES_MOR_ST(I) > MAIN_RES_MOR_ST(141) -
3285*43560;  !3285 acre-feet is 1 feet of storage in Moore;
[MAIN_RES_MOR_ST_TARG_DD_BASS_L2] MAIN_RES_MOR_ST(I) < MAIN_RES_MOR_ST(141) +
3285*43560;  !3285 acre-feet is 1 feet of storage in Moore;
[MAIN_RES_COM_ST_TARG_DD_BASS_L1] MAIN_RES_COM_ST(I) > MAIN_RES_COM_ST(141) -
1025*43560;  !1025 acre-feet is 1 feet of storage in Comerford;
[MAIN_RES_COM_ST_TARG_DD_BASS_L2] MAIN_RES_COM_ST(I) < MAIN_RES_COM_ST(141) +
1025*43560;  !1025 acre-feet is 1 feet of storage in Comerford;
);

!Deerfield;
[DRF_RES_SOM_ST_TARG_LOON_L]  DRF_RES_SOM_ST(121) > 46300*43560;          !Day 121 is
May 1;
@FOR(day(I)| I #GE# 122 #AND# I #LE# 212:                                  !Day 122 is
May 2, Day 212 is July 31;
[DRF_RES_SOM_ST_TARG_LOON_DD_L1] DRF_RES_SOM_ST(I) > DRF_RES_SOM_ST(121) - 4090*43560;
!4090 acre-feet is 3 feet of storage in Somerset;
[DRF_RES_SOM_ST_TARG_LOON_DD_L2] DRF_RES_SOM_ST(I) < DRF_RES_SOM_ST(121) + 4090*43560;
!4090 acre-feet is 3 feet of storage in Somerset;
);


[DRF_RES_HAR_ST_TARG_FISHSPAWN_L]  DRF_RES_HAR_ST(91) > 107520*43560;  !Day 91 is
April 1;
@FOR(day(I)| I #GE# 92 #AND# I #LE# 166:                                   !Day 92 is
April 2, Day 166 is June 15;
[DRF_RES_HAR_ST_TARG_FISHSPAWN_DD_L1]  DRF_RES_HAR_ST(I) >  DRF_RES_HAR_ST(91);
);

@FOR(day(I)| I #GE# 167 #AND# I #LE# 196:                          !Day 167 is June 16,
Day 196 is July 15;
[DRF_RES_HAR_SUMMER_DD] DRF_RES_HAR_ST(I-1)-DRF_RES_HAR_ST(I) <
DRF_RES_HAR_SUMMER_RAMP;
);


!Release Constraints and Targets - Each reservoir has a minimum release target and
some have a maximum release target;
@FOR (day(I) | I #GE# 3:
!Upper Third;
```

```
[UCN_RES_SCL_R_TARG_MIN_L] UCN_RES_SCL_R(I) > UCN_RES_SCL_R_TARG_MIN(I);
[UCN_RES_FCL_R_TARG_MIN_L] UCN_RES_FCL_R(I) > UCN_RES_FCL_R_TARG_MIN(I);
[UCN_RES_LFR_R_TARG_MIN_L] UCN_RES_LFR_R(I) > UCN_RES_LFR_R_TARG_MIN(I);
[MAIN_RES_MOR_R_TARG_MIN_L] MAIN_RES_MOR_R(I) > MAIN_RES_MOR_R_TARG_MIN(I);
[MAIN_RES_COM_R_TARG_MIN_L] MAIN_RES_COM_R(I) > MAIN_RES_COM_R_TARG_MIN(I);
[MAIN_RES_MCD_R_TARG_MIN_L] MAIN_RES_MCD_R(I) > MAIN_RES_MCD_R_TARG_MIN(I);

!Middle Third;
[BLK_RES_NSP_R_TARG_MIN_L] BLK_RES_NSP_R(I) > BLK_RES_NSP_R_TARG_MIN(I);
[MAS_RES_GOO_R_TARG_MIN_L] MAS_RES_GOO_R(I) > MAS_RES_GOO_R_TARG_MIN(I);
[MAS_RES_CRY_R_TARG_MIN_L] MAS_RES_CRY_R(I) > MAS_RES_CRY_R_TARG_MIN(I);
[MAS_RES_GRF_R_TARG_MIN_L] MAS_RES_GRF_R(I) > MAS_RES_GRF_R_TARG_MIN(I);
[MAS_RES_MSL_R_TARG_MIN_L] MAS_RES_MSL_R(I) > MAS_RES_MSL_R_TARG_MIN(I);
[OMP_RES_UNV_R_TARG_MIN_L] OMP_RES_UNV_R(I) > OMP_RES_UNV_R_TARG_MIN(I);
[OTT_RES_NHD_R_TARG_MIN_L] OTT_RES_NHD_R(I) > OTT_RES_NHD_R_TARG_MIN(I);
[SUG_RES_LSU_R_TARG_MIN_L] SUG_RES_LSU_R(I) > SUG_RES_LSU_R_TARG_MIN(I);
[MAIN_RES_BFA_R_TARG_MIN_L] MAIN_RES_BFA_R(I) > MAIN_RES_BFA_R_TARG_MIN(I);
[MAIN_RES_WLD_R_TARG_MIN_L] MAIN_RES_WLD_R(I) > MAIN_RES_WLD_R_TARG_MIN(I);
[OMP_RES_UNV_R_TARG_MAX_L] OMP_RES_UNV_R(I) < OMP_RES_UNV_R_TARG_MAX;
[OTT_RES_NHD_R_TARG_MAX_L] OTT_RES_NHD_R(I) < OTT_RES_NHD_R_TARG_MAX;
[BLK_RES_NSP_R_TARG_MAX_L] BLK_RES_NSP_R(I) < BLK_RES_NSP_R_TARG_MAX;

!West;
[WST_RES_BMD_R_TARG_MAX_L] WST_RES_BMD_R(I) < WST_RES_BMD_R_TARG_MAX;
[WST_RES_BMD_R_TARG_MIN_L] WST_RES_BMD_R(I) > WST_RES_BMD_R_TARG_MIN(I);
[WST_RES_TWN_R_TARG_MAX_L] WST_RES_TWN_R(I) < WST_RES_TWN_R_TARG_MAX;
[WST_RES_TWN_R_TARG_MIN_L] WST_RES_TWN_R(I) > WST_RES_TWN_R_TARG_MIN(I);

!Ashuelot;
[ASH_RES_SMD_R_TARG_MAX_L] ASH_RES_SMD_R(I) < ASH_RES_SMD_R_TARG_MAX(I);
[ASH_RES_SMD_R_TARG_MIN_L] ASH_RES_SMD_R(I) > ASH_RES_SMD_R_TARG_MIN(I);
[ASH_RES_OBD_R_TARG_MAX_L] ASH_RES_OBD_R(I) < ASH_RES_OBD_R_TARG_MAX;
[ASH_RES_OBD_R_TARG_MIN_L] ASH_RES_OBD_R(I) > ASH_RES_OBD_R_TARG_MIN(I);

!Millers;
[MLR_RES_BIR_R_TARG_MIN_L] MLR_RES_BIR_R > MLR_RES_BIR_R_TARG_MIN(I);
[MLR_RES_TUL_R_TARG_MIN_L] MLR_RES_TUL_R > MLR_RES_TUL_R_TARG_MIN(I);
[MLR_RES_BIR_R_TARG_MAX_L] MLR_RES_BIR_R < MLR_RES_BIR_R_TARG_MAX;
[MLR_RES_TUL_R_TARG_MAX_L] MLR_RES_TUL_R < MLR_RES_TUL_R_TARG_MAX;

!Deerfield;
[DRF_RES_SOM_R_TARG_MIN_L] DRF_RES_SOM_R(I) > DRF_RES_SOM_R_TARG_MIN(I);
[DRF_RES_SBG_R_TARG_MIN_L] DRF_RES_SBG_R(I) > DRF_RES_SBG_R_TARG_MIN(I);
[DRF_RES_SBG_DR_TARG_MIN_L] DRF_RES_SBG_DR(I) > DRF_RES_SBG_DR_TARG_MIN(I);
[DRF_RES_HAR_DR_TARG_MIN_L] DRF_RES_HAR_DR(I) > DRF_RES_HAR_DR_TARG_MIN(I);
[DRF_RES_SHR_R_TARG_MIN_L] DRF_RES_SHR_R(I) > DRF_RES_SHR_R_TARG_MIN(I);
[DRF_RES_DV5_DR_TARG_MIN_L] DRF_RES_DV5_DR(I) > DRF_RES_DV5_DR_TARG_MIN(I);
[DRF_RES_FBR_R_TARG_MIN_L] DRF_RES_FBR_R(I) > DRF_RES_FBR_R_TARG_MIN(I);
[DRF_RES_DV4_R_TARG_MIN_L] DRF_RES_DV4_R(I) > DRF_RES_DV4_R_TARG_MIN(I);
[DRF_RES_DV3_R_TARG_MIN_L] DRF_RES_DV3_R(I) > DRF_RES_DV3_R_TARG_MIN(I);
[DRF_RES_GRD_DR_TARG_MIN_L] DRF_RES_GRD_DR(I) > DRF_RES_GRD_DR_TARG_MIN(I);
[DRF_RES_DV2_R_TARG_MIN_L] DRF_RES_DV2_R(I) > DRF_RES_DV2_R_TARG_MIN(I);
[DRF_RES_SOM_R_TARG_MAX_L] DRF_RES_SOM_R(I)< DRF_RES_SOM_R_TARG_MAX(I);

!Chicopee;
[CHP_RES_BFD_R_TARG_MIN_L] CHP_RES_BFD_R(I) > CHP_RES_BFD_R_TARG_MIN(I);    !Barre
Falls Minimum Release Target;
[CHP_RES_CBD_R_TARG_MIN_L] CHP_RES_CBD_R(I) > CHP_RES_CBD_R_TARG_MIN(I);    !Conant
Brook Minimum Release Target;
[CHP_RES_QWD_R_TARG_MIN_L] CHP_RES_QWD_R(I) > CHP_RES_QWD_R_TARG_MIN(I);    !Quabbin
Minimum Release Target;
[CHP_RES_BFD_R_TARG_MAX_L] CHP_RES_BFD_R(I) < CHP_RES_BFD_R_TARG_MAX;    !Barre Falls
Maximum Release Target;
```

```
[CHP_RES_CBD_R_TARG_MAX_L] CHP_RES_CBD_R(I) < CHP_RES_CBD_R_TARG_MAX;    !Conant Brook
Maximum Release Target;
[CHP_RES_QWD_R_TARG_MAX_L] CHP_RES_QWD_R(I) - CHP_RES_QWD_R_TARG_MAX(I) =
CHP_RES_QWD_R_TARG_MAX_ABOVE(I) - CHP_RES_QWD_R_TARG_MAX_BELOW(I);    !Quabbin Maximum
Release Target;

!Westfield;
[WSF_RES_KVL_R_TARG_MIN_L] WSF_RES_KVL_R(I) > WSF_RES_KVL_R_TARG_MIN(I);
[WSF_RES_LVL_R_TARG_MIN_L] WSF_RES_LVL_R(I) > WSF_RES_LVL_R_TARG_MIN(I);
[WSF_RES_KVL_R_TARG_MAX_L] WSF_RES_KVL_R(I) < WSF_RES_KVL_R_TARG_MAX;
[WSF_RES_LVL_R_TARG_MAX_L] WSF_RES_LVL_R(I) < WSF_RES_LVL_R_TARG_MAX;
[WSF_RES_BBK_R_TARG_MAX_L] WSF_RES_BBK_R(I) < WSF_RES_BBK_R_TARG_MAX;
[WSF_RES_BBK_R_TARG_MIN_L] WSF_RES_BBK_R(I) = WSF_RES_BBK_Q(I);
[WSF_RES_CMT_R_TARG_MAX_L] WSF_RES_CMT_R(I) < WSF_RES_CMT_R_TARG_MAX;

!Farmington;
[FAR_RES_OTI_R_TARG_MIN_L] FAR_RES_OTI_R(I) > FAR_RES_OTI_R_TARG_MIN(I);
[FAR_RES_WBR_R_TARG_MIN_L] FAR_RES_WBR_R(I) - FAR_RES_WBR_R_TARG_MIN(I) =
FAR_RES_WBR_R_TARG_MIN_ABOVE(I) - FAR_RES_WBR_R_TARG_MIN_BELOW(I);
[FAR_RES_COL_R_TARG_MIN_L] FAR_RES_COL_R(I) - FAR_RES_COL_R_TARG_MIN(I) =
FAR_RES_COL_R_TARG_MIN_ABOVE(I) - FAR_RES_COL_R_TARG_MIN_BELOW(I);
[FAR_RES_OTI_R_TARG_MAX_L] FAR_RES_OTI_R(I) - FAR_RES_OTI_R_TARG_MAX(I) =
FAR_RES_OTI_R_TARG_MAX_ABOVE(I) - FAR_RES_OTI_R_TARG_MAX_BELOW(I);
[FAR_RES_COL_R_TARG_MAX_L] FAR_RES_COL_R(I) < FAR_RES_COL_R_TARG_MAX;
![FAR_RES_WBR_R_TARG_MAX_L] FAR_RES_WBR_R(I) < FAR_RES_WBR_R_TARG_MAX;

!Lower Third Mainstem;
[MAIN_RES_VRN_R_TARG_MIN_L] MAIN_RES_VRN_R(I) > MAIN_RES_VRN_R_TARG_MIN(I);
[MAIN_RES_TRN_DR_TARG_MIN_L] MAIN_RES_TRN_DR(I) > MAIN_RES_TRN_DR_TARG_MIN(I);
[MAIN_RES_HOL_DR_TARG_MIN_L] MAIN_RES_HOL_DR(I) > MAIN_RES_HOL_DR_TARG_MIN(I);
);
```

Ramping Rates

Ramping rates are designed to force modeled storage operations to remain within normal values and prevent erratic output and in some instances mimic pre-defined storage ramping rules for specific reservoirs.

```
!Ramping Up Releases;
@FOR (day(I) | I #GE# 2:

!Upper Third;
[UCN_RES_SCL_RAMP_UP] UCN_RES_SCL_ST(I-1) - UCN_RES_SCL_ST(I) < UCN_RES_SCL_RAMP;
[UCN_RES_FCL_RAMP_UP] UCN_RES_FCL_ST(I-1) - UCN_RES_FCL_ST(I) < UCN_RES_FCL_RAMP;
[UCN_RES_LFR_RAMP_UP] UCN_RES_LFR_ST(I-1) - UCN_RES_LFR_ST(I) < UCN_RES_LFR_RAMP;
[MAIN_RES_MOR_RAMP_UP] MAIN_RES_MOR_ST(I-1) - MAIN_RES_MOR_ST(I) < MAIN_RES_MOR_RAMP;
[MAIN_RES_COM_RAMP_UP] MAIN_RES_COM_ST(I-1) - MAIN_RES_COM_ST(I) < MAIN_RES_COM_RAMP;
[MAIN_RES_MCD_RAMP_UP] MAIN_RES_MCD_ST(I-1) - MAIN_RES_MCD_ST(I) < MAIN_RES_MCD_RAMP;

!Middle Third;
[OMP_RES_UNV_RAMP_UP1] OMP_RES_UNV_ST(I-1) - OMP_RES_UNV_ST(I) < OMP_RES_UNV_RAMP;
[OTT_RES_NHD_RAMP_UP1] OTT_RES_NHD_ST(I-1) - OTT_RES_NHD_ST(I) < OTT_RES_NHD_RAMP;
[BLK_RES_NSP_RAMP_UP1] BLK_RES_NSP_ST(I-1) - BLK_RES_NSP_ST(I) < BLK_RES_NSP_RAMP;

!West;
[WST_RES_BMD_RAMP_UP1] WST_RES_BMD_ST(I-1) - WST_RES_BMD_ST(I) < WST_RES_BMD_RAMP;
[WST_RES_TWN_RAMP_UP1] WST_RES_TWN_ST(I-1) - WST_RES_TWN_ST(I) < WST_RES_TWN_RAMP;

!Ashuelot;
[ASH_RES_SMD_RAMP_UP1] ASH_RES_SMD_ST(I-1) - ASH_RES_SMD_ST(I) < ASH_RES_SMD_RAMP;
[ASH_RES_OBD_RAMP_UP1] ASH_RES_OBD_ST(I-1) - ASH_RES_OBD_ST(I) < ASH_RES_OBD_RAMP;
```

```
!Millers;
[MLR_RES_TUL_RAMP_UP1] MLR_RES_TUL_ST(I-1) - MLR_RES_TUL_ST(I) < MLR_RES_TUL_RAMP;
[MLR_RES_BIR_RAMP_UP1] MLR_RES_BIR_ST(I-1) - MLR_RES_BIR_ST(I) < MLR_RES_BIR_RAMP;

!Deerfield;
[DRF_RES_SOM_RAMP_UP] DRF_RES_SOM_ST(I-1)-DRF_RES_SOM_ST(I) < DRF_RES_SOM_RAMP;
[DRF_RES_HAR_RAMP_UP] DRF_RES_HAR_ST(I-1)-DRF_RES_HAR_ST(I) < DRF_RES_HAR_RAMP;

!Chicoppe;
[CHP_RES_BFD_RAMP_UP1] CHP_RES_BFD_ST(I-1) - CHP_RES_BFD_ST(I) < CHP_RES_BFD_RAMP;
[CHP_RES_CBD_RAMP_UP1] CHP_RES_CBD_ST(I-1) - CHP_RES_CBD_ST(I) < CHP_RES_CBD_RAMP;

!Westfield;
[WSF_RES_KVL_RAMP_UP1] WSF_RES_KVL_ST(I-1) - WSF_RES_KVL_ST(I) < WSF_RES_KVL_RAMP;
[WSF_RES_LVL_RAMP_UP1] WSF_RES_LVL_ST(I-1) - WSF_RES_LVL_ST(I) < WSF_RES_LVL_RAMP;

!Farmington;
[FAR_RES_COL_RAMP_UP1] FAR_RES_COL_ST(I-1) - FAR_RES_COL_ST(I) < FAR_RES_COL_RAMP;
);
```

Initial/Final Storage Conditions

Since the model runs on a daily time step for a year at a time, the initial and final storage conditions are required to remain equal at each modeled location. In this way, reservoir storage levels on December 31, 2003 and January 1, 2004 do not vary drastically and generate erratic output.

```
!Other Constraints;
!Contain the First and Last day's storage to equal the target on that day;
NUMDAYS = @SIZE(day);

!Upper Third;
[UCN_RES_SCL_ST_START] UCN_RES_SCL_ST(1) = UCN_RES_SCL_ST_INT;!Second CT Lake;
[UCN_RES_SCL_ST_START2] UCN_RES_SCL_ST(2) = UCN_RES_SCL_ST_INT;!Second CT Lake;
[UCN_RES_SCL_ST_END1] UCN_RES_SCL_ST(NUMDAYS) - UCN_RES_SCL_ST_END =
UCN_RES_SCL_ST_END_ABOVE - UCN_RES_SCL_ST_END_BELOW; !Second CT Lake;

[UCN_RES_FCL_ST_START] UCN_RES_FCL_ST(1) = UCN_RES_FCL_ST_INT;!First CT Lake;
[UCN_RES_FCL_ST_START2] UCN_RES_FCL_ST(2) = UCN_RES_FCL_ST_INT;!First CT Lake;
[UCN_RES_FCL_ST_END1] UCN_RES_FCL_ST(NUMDAYS) - UCN_RES_FCL_ST_END =
UCN_RES_FCL_ST_END_ABOVE - UCN_RES_FCL_ST_END_BELOW;

[UCN_RES_LFR_ST_START] UCN_RES_LFR_ST(1) = UCN_RES_LFR_ST_INT;!Lake Francis;
[UCN_RES_LFR_ST_START2] UCN_RES_LFR_ST(2) = UCN_RES_LFR_ST_INT;!Lake Francis;
[UCN_RES_LFR_ST_END1] UCN_RES_LFR_ST(NUMDAYS) - UCN_RES_LFR_ST_END =
UCN_RES_LFR_ST_END_ABOVE - UCN_RES_LFR_ST_END_BELOW;

[MAIN_RES_MOR_ST_START] MAIN_RES_MOR_ST(1) = MAIN_RES_MOR_ST_INT;!Moore;
[MAIN_RES_MOR_ST_START2] MAIN_RES_MOR_ST(2) = MAIN_RES_MOR_ST_INT;!Moore;
[MAIN_RES_MOR_ST_END1] MAIN_RES_MOR_ST(NUMDAYS) - MAIN_RES_MOR_ST_END =
MAIN_RES_MOR_ST_END_ABOVE - MAIN_RES_MOR_ST_END_BELOW;

[MAIN_RES_COM_ST_START] MAIN_RES_COM_ST(1) = MAIN_RES_COM_ST_INT;!Comerford;
[MAIN_RES_COM_ST_START2] MAIN_RES_COM_ST(2) = MAIN_RES_COM_ST_INT;!Comerford;
[MAIN_RES_COM_ST_END1] MAIN_RES_COM_ST(NUMDAYS) - MAIN_RES_COM_ST_END =
MAIN_RES_COM_ST_END_ABOVE - MAIN_RES_COM_ST_END_BELOW;

[MAIN_RES_MCD_ST_START] MAIN_RES_MCD_ST(1) = MAIN_RES_MCD_ST_INT;!MCD;
[MAIN_RES_MCD_ST_START2] MAIN_RES_MCD_ST(2) = MAIN_RES_MCD_ST_INT;!MCD;
[MAIN_RES_MCD_ST_END1] MAIN_RES_MCD_ST(NUMDAYS) - MAIN_RES_MCD_ST_END =
MAIN_RES_MCD_ST_END_ABOVE - MAIN_RES_MCD_ST_END_BELOW;
```

```
!Middle Third;
[BLK_RES_NSP_ST_START] BLK_RES_NSP_ST(1) = BLK_RES_NSP_ST_INT;
[BLK_RES_NSP_ST_START2] BLK_RES_NSP_ST(2) = BLK_RES_NSP_ST_INT;
[BLK_RES_NSP_ST_END1] BLK_RES_NSP_ST(NUMDAYS) - BLK_RES_NSP_ST_END =
BLK_RES_NSP_ST_END_ABOVE - BLK_RES_NSP_ST_END_BELOW;

[MAS_RES_GOO_ST_START] MAS_RES_GOO_ST(1) = MAS_RES_GOO_ST_INT;
[MAS_RES_GOO_ST_START2] MAS_RES_GOO_ST(2) = MAS_RES_GOO_ST_INT;
[MAS_RES_GOO_ST_END1] MAS_RES_GOO_ST(NUMDAYS) - MAS_RES_GOO_ST_END =
MAS_RES_GOO_ST_END_ABOVE - MAS_RES_GOO_ST_END_BELOW;

[MAS_RES_CRY_ST_START] MAS_RES_CRY_ST(1) = MAS_RES_CRY_ST_INT;
[MAS_RES_CRY_ST_START2] MAS_RES_CRY_ST(2) = MAS_RES_CRY_ST_INT;
[MAS_RES_CRY_ST_END1] MAS_RES_CRY_ST(NUMDAYS) - MAS_RES_CRY_ST_END =
MAS_RES_CRY_ST_END_ABOVE - MAS_RES_CRY_ST_END_BELOW;

[MAS_RES_GRF_ST_START] MAS_RES_GRF_ST(1) = MAS_RES_GRF_ST_INT;
[MAS_RES_GRF_ST_START2] MAS_RES_GRF_ST(2) = MAS_RES_GRF_ST_INT;
[MAS_RES_GRF_ST_END1] MAS_RES_GRF_ST(NUMDAYS) - MAS_RES_GRF_ST_END =
MAS_RES_GRF_ST_END_ABOVE - MAS_RES_GRF_ST_END_BELOW;

[MAS_RES_MSL_ST_START] MAS_RES_MSL_ST(1) = MAS_RES_MSL_ST_INT;
[MAS_RES_MSL_ST_START2] MAS_RES_MSL_ST(2) = MAS_RES_MSL_ST_INT;
[MAS_RES_MLS_ST_END1] MAS_RES_MSL_ST(NUMDAYS) - MAS_RES_MSL_ST_END =
MAS_RES_MSL_ST_END_ABOVE - MAS_RES_MSL_ST_END_BELOW;

[OMP_RES_UNV_ST_START] OMP_RES_UNV_ST(1) = OMP_RES_UNV_ST_INT;
[OMP_RES_UNV_ST_START2] OMP_RES_UNV_ST(2) = OMP_RES_UNV_ST_INT;
[OMP_RES_UNV_ST_END1] OMP_RES_UNV_ST(NUMDAYS) - OMP_RES_UNV_ST_END =
OMP_RES_UNV_ST_END_ABOVE - OMP_RES_UNV_ST_END_BELOW;



[OTT_RES_NHD_ST_START] OTT_RES_NHD_ST(1) = OTT_RES_NHD_ST_INT;
[OTT_RES_NHD_ST_START2] OTT_RES_NHD_ST(2) = OTT_RES_NHD_ST_INT;
[OTT_RES_NHD_ST_END1] OTT_RES_NHD_ST(NUMDAYS) - OTT_RES_NHD_ST_END =
OTT_RES_NHD_ST_END_ABOVE - OTT_RES_NHD_ST_END_BELOW;

[SUG_RES_LSU_ST_START] SUG_RES_LSU_ST(1) = SUG_RES_LSU_ST_INT;
[SUG_RES_LSU_ST_START2] SUG_RES_LSU_ST(2) = SUG_RES_LSU_ST_INT;
[SUG_RES_LSU_ST_END1] SUG_RES_LSU_ST(NUMDAYS) - SUG_RES_LSU_ST_END =
SUG_RES_LSU_ST_END_ABOVE - SUG_RES_LSU_ST_END_BELOW;

[SUG_RES_SGR_ST_START] SUG_RES_SGR_ST(1) = SUG_RES_SGR_ST_INT;
[SUG_RES_SGR_ST_START2] SUG_RES_SGR_ST(2) = SUG_RES_SGR_ST_INT;
[SUG_RES_SGR_ST_END1] SUG_RES_SGR_ST(NUMDAYS) - SUG_RES_SGR_ST_END =
SUG_RES_SGR_ST_END_ABOVE - SUG_RES_SGR_ST_END_BELOW;

[MAIN_RES_BFA_ST_START] MAIN_RES_BFA_ST(1) = MAIN_RES_BFA_ST_INT;
[MAIN_RES_BFA_ST_START2] MAIN_RES_BFA_ST(2) = MAIN_RES_BFA_ST_INT;
[MAIN_RES_BFA_ST_END1] MAIN_RES_BFA_ST(NUMDAYS) - MAIN_RES_BFA_ST_END =
MAIN_RES_BFA_ST_END_ABOVE - MAIN_RES_BFA_ST_END_BELOW;

[MAIN_RES_WLD_ST_START] MAIN_RES_WLD_ST(1) = MAIN_RES_WLD_ST_INT;
[MAIN_RES_WLD_ST_START2] MAIN_RES_WLD_ST(2) = MAIN_RES_WLD_ST_INT;
[MAIN_RES_WLD_ST_END1] MAIN_RES_WLD_ST(NUMDAYS) - MAIN_RES_WLD_ST_END =
MAIN_RES_WLD_ST_END_ABOVE - MAIN_RES_WLD_ST_END_BELOW;

!West;
[WST_RES_BMD_ST_START] WST_RES_BMD_ST(1) = WST_RES_BMD_ST_INT;
[WST_RES_BMD_ST_START2] WST_RES_BMD_ST(2) = WST_RES_BMD_ST_INT;
```

```
[WST_RES_BMD_ST_END1] WST_RES_BMD_ST(NUMDAYS) - WST_RES_BMD_ST_END =
WST_RES_BMD_ST_END_ABOVE - WST_RES_BMD_ST_END_BELOW;

[WST_RES_TWN_ST_START] WST_RES_TWN_ST(1) = WST_RES_TWN_ST_INT;
[WST_RES_TWN_ST_START2] WST_RES_TWN_ST(2) = WST_RES_TWN_ST_INT;
[WST_RES_TWN_ST_END1] WST_RES_TWN_ST(NUMDAYS) - WST_RES_TWN_ST_END =
WST_RES_TWN_ST_END_ABOVE - WST_RES_TWN_ST_END_BELOW;

!Ashuelot;
[ASH_RES_SMD_ST_START] ASH_RES_SMD_ST(1) = ASH_RES_SMD_ST_INT;
[ASH_RES_SMD_ST_START2] ASH_RES_SMD_ST(2) = ASH_RES_SMD_ST_INT;
[ASH_RES_SMD_ST_END1] ASH_RES_SMD_ST(NUMDAYS) - ASH_RES_SMD_ST_END =
ASH_RES_SMD_ST_END_ABOVE - ASH_RES_SMD_ST_END_BELOW;

[ASH_RES_OBD_ST_START] ASH_RES_OBD_ST(1) = ASH_RES_OBD_ST_INT;
[ASH_RES_OBD_ST_START2] ASH_RES_OBD_ST(2) = ASH_RES_OBD_ST_INT;
[ASH_RES_OBD_ST_END1] ASH_RES_OBD_ST(NUMDAYS) - ASH_RES_OBD_ST_END =
ASH_RES_OBD_ST_END_ABOVE - ASH_RES_OBD_ST_END_BELOW;

!Millers;
[MLR_RES_MON_ST_START] MLR_RES_MON_ST(1) = MLR_RES_MON_ST_INT;
[MLR_RES_MON_ST_START2] MLR_RES_MON_ST(2) = MLR_RES_MON_ST_INT;
[MLR_RES_MON_ST_END1] MLR_RES_MON_ST(NUMDAYS) - MLR_RES_MON_ST_END =
MLR_RES_MON_ST_END_ABOVE - MLR_RES_MON_ST_END_BELOW;

[MLR_RES_NEK_ST_START] MLR_RES_NEK_ST(1) = MLR_RES_NEK_ST_INT;
[MLR_RES_NEK_ST_START2] MLR_RES_NEK_ST(2) = MLR_RES_NEK_ST_INT;
[MLR_RES_NEK_ST_END1] MLR_RES_NEK_ST(NUMDAYS) - MLR_RES_NEK_ST_END =
MLR_RES_NEK_ST_END_ABOVE - MLR_RES_NEK_ST_END_BELOW;

[MLR_RES_BIR_ST_START] MLR_RES_BIR_ST(1) = MLR_RES_BIR_ST_INT;
[MLR_RES_BIR_ST_START2] MLR_RES_BIR_ST(2) = MLR_RES_BIR_ST_INT;
[MLR_RES_BIR_ST_END1] MLR_RES_BIR_ST(NUMDAYS) - MLR_RES_BIR_ST_END =
MLR_RES_BIR_ST_END_ABOVE - MLR_RES_BIR_ST_END_BELOW;

[MLR_RES_TUL_ST_START] MLR_RES_TUL_ST(1) = MLR_RES_TUL_ST_INT;
[MLR_RES_TUL_ST_START2] MLR_RES_TUL_ST(2) = MLR_RES_TUL_ST_INT;
[MLR_RES_TUL_ST_END1] MLR_RES_TUL_ST(NUMDAYS) - MLR_RES_TUL_ST_END =
MLR_RES_TUL_ST_END_ABOVE - MLR_RES_TUL_ST_END_BELOW;

!Deerfield;
[DRF_RES_SOM_ST_START] DRF_RES_SOM_ST(1) = DRF_RES_SOM_ST_INT;
[DRF_RES_SOM_ST_START2] DRF_RES_SOM_ST(2) = DRF_RES_SOM_ST_INT;
[DRF_RES_SOM_ST_END1] DRF_RES_SOM_ST(NUMDAYS) - DRF_RES_SOM_ST_END =
DRF_RES_SOM_ST_END_ABOVE - DRF_RES_SOM_ST_END_BELOW;

[DRF_RES_SBG_ST_START] DRF_RES_SBG_ST(1) = DRF_RES_SBG_ST_INT;
[DRF_RES_SBG_ST_START2] DRF_RES_SBG_ST(2) = DRF_RES_SBG_ST_INT;
[DRF_RES_SBG_ST_END1] DRF_RES_SBG_ST(NUMDAYS) - DRF_RES_SBG_ST_END =
DRF_RES_SBG_ST_END_ABOVE - DRF_RES_SBG_ST_END_BELOW;

[DRF_RES_HAR_ST_START] DRF_RES_HAR_ST(1) = DRF_RES_HAR_ST_INT;
[DRF_RES_HAR_ST_START2] DRF_RES_HAR_ST(2) = DRF_RES_HAR_ST_INT;
[DRF_RES_HAR_ST_END1] DRF_RES_HAR_ST(NUMDAYS) - DRF_RES_HAR_ST_END =
DRF_RES_HAR_ST_END_ABOVE - DRF_RES_HAR_ST_END_BELOW;

[DRF_RES_SHR_ST_START] DRF_RES_SHR_ST(1) = DRF_RES_SHR_ST_INT;
[DRF_RES_SHR_ST_START2] DRF_RES_SHR_ST(2) = DRF_RES_SHR_ST_INT;
[DRF_RES_SHR_ST_END1] DRF_RES_SHR_ST(NUMDAYS) - DRF_RES_SHR_ST_END =
DRF_RES_SHR_ST_END_ABOVE - DRF_RES_SHR_ST_END_BELOW;

[DRF_RES_FBR_ST_START] DRF_RES_FBR_ST(1) = DRF_RES_FBR_ST_INT;
[DRF_RES_FBR_ST_START2] DRF_RES_FBR_ST(2) = DRF_RES_FBR_ST_INT;
```

```
[DRF_RES_FBR_ST_END1] DRF_RES_FBR_ST(NUMDAYS) - DRF_RES_FBR_ST_END =
DRF_RES_FBR_ST_END_ABOVE - DRF_RES_FBR_ST_END_BELOW;

[DRF_RES_DV5_ST_START] DRF_RES_DV5_ST(1) = DRF_RES_DV5_ST_INT;
[DRF_RES_DV5_ST_START2] DRF_RES_DV5_ST(2) = DRF_RES_DV5_ST_INT;
[DRF_RES_DV5_ST_END1] DRF_RES_DV5_ST(NUMDAYS) - DRF_RES_DV5_ST_END =
DRF_RES_DV5_ST_END_ABOVE - DRF_RES_DV5_ST_END_BELOW;

[DRF_RES_DV4_ST_START] DRF_RES_DV4_ST(1) = DRF_RES_DV4_ST_INT;
[DRF_RES_DV4_ST_START2] DRF_RES_DV4_ST(2) = DRF_RES_DV4_ST_INT;
[DRF_RES_DV4_ST_END1] DRF_RES_DV4_ST(NUMDAYS) - DRF_RES_DV4_ST_END =
DRF_RES_DV4_ST_END_ABOVE - DRF_RES_DV4_ST_END_BELOW;

[DRF_RES_DV3_ST_START] DRF_RES_DV3_ST(1) = DRF_RES_DV3_ST_INT;
[DRF_RES_DV3_ST_START2] DRF_RES_DV3_ST(2) = DRF_RES_DV3_ST_INT;
[DRF_RES_DV3_ST_END1] DRF_RES_DV3_ST(NUMDAYS) - DRF_RES_DV3_ST_END =
DRF_RES_DV3_ST_END_ABOVE - DRF_RES_DV3_ST_END_BELOW;

[DRF_RES_GRD_ST_START] DRF_RES_GRD_ST(1) = DRF_RES_GRD_ST_INT;
[DRF_RES_GRD_ST_START2] DRF_RES_GRD_ST(2) = DRF_RES_GRD_ST_INT;
[DRF_RES_GRD_ST_END1] DRF_RES_GRD_ST(NUMDAYS) - DRF_RES_GRD_ST_END =
DRF_RES_GRD_ST_END_ABOVE - DRF_RES_GRD_ST_END_BELOW;

[DRF_RES_DV2_ST_START] DRF_RES_DV2_ST(1) = DRF_RES_DV2_ST_INT;
[DRF_RES_DV2_ST_START2] DRF_RES_DV2_ST(2) = DRF_RES_DV2_ST_INT;
[DRF_RES_DV2_ST_END1] DRF_RES_DV2_ST(NUMDAYS) - DRF_RES_DV2_ST_END =
DRF_RES_DV2_ST_END_ABOVE - DRF_RES_DV2_ST_END_BELOW;

!Chicopee;
[CHP_RES_BFD_ST_START] CHP_RES_BFD_ST(1) = CHP_RES_BFD_ST_INT;
[CHP_RES_BFD_ST_START2] CHP_RES_BFD_ST(2) = CHP_RES_BFD_ST_INT;
[CHP_RES_BFD_ST_END1] CHP_RES_BFD_ST(NUMDAYS) - CHP_RES_BFD_ST_END =
CHP_RES_BFD_ST_END_ABOVE - CHP_RES_BFD_ST_END_BELOW;

[CHP_RES_CBD_ST_START] CHP_RES_CBD_ST(1) = CHP_RES_CBD_ST_INT;
[CHP_RES_CBD_ST_START2] CHP_RES_CBD_ST(2) = CHP_RES_CBD_ST_INT;
[CHP_RES_CBD_ST_END1] CHP_RES_CBD_ST(NUMDAYS) - CHP_RES_CBD_ST_END =
CHP_RES_CBD_ST_END_ABOVE - CHP_RES_CBD_ST_END_BELOW;

[CHP_RES_QWD_ST_START] CHP_RES_QWD_ST(1) = CHP_RES_QWD_ST_INT;
[CHP_RES_QWD_ST_START2] CHP_RES_QWD_ST(2) = CHP_RES_QWD_ST_INT;
[CHP_RES_QWD_ST_END1] CHP_RES_QWD_ST(NUMDAYS) - CHP_RES_QWD_ST_END =
CHP_RES_QWD_ST_END_ABOVE - CHP_RES_QWD_ST_END_BELOW;

[CHP_RES_MMBR_ST_START] CHP_RES_MMBR_ST(1) = CHP_RES_MMBR_ST_INT;
[CHP_RES_MMBR_ST_START2] CHP_RES_MMBR_ST(2) = CHP_RES_MMBR_ST_INT;
[CHP_RES_MMBR_ST_END1] CHP_RES_MMBR_ST(NUMDAYS) - CHP_RES_MMBR_ST_END =
CHP_RES_MMBR_ST_END_ABOVE - CHP_RES_MMBR_ST_END_BELOW;

[CHP_RES_RBD_ST_START] CHP_RES_RBD_ST(1) = CHP_RES_RBD_ST_INT;
[CHP_RES_RBD_ST_START2] CHP_RES_RBD_ST(2) = CHP_RES_RBD_ST_INT;
[CHP_RES_RBD_ST_END1] CHP_RES_RBD_ST(NUMDAYS) - CHP_RES_RBD_ST_END =
CHP_RES_RBD_ST_END_ABOVE - CHP_RES_RBD_ST_END_BELOW;

!Westfield;
[WSF_RES_KVL_ST_START] WSF_RES_KVL_ST(1) = WSF_RES_KVL_ST_INT;
[WSF_RES_KVL_ST_START2] WSF_RES_KVL_ST(2) = WSF_RES_KVL_ST_INT;
[WSF_RES_KVL_ST_END1] WSF_RES_KVL_ST(NUMDAYS) - WSF_RES_KVL_ST_END =
WSF_RES_KVL_ST_END_ABOVE - WSF_RES_KVL_ST_END_BELOW;

[WSF_RES_LVL_ST_START] WSF_RES_LVL_ST(1) = WSF_RES_LVL_ST_INT;
[WSF_RES_LVL_ST_START2] WSF_RES_LVL_ST(2) = WSF_RES_LVL_ST_INT;
```

```
[WSF_RES_LVL_ST_END1] WSF_RES_LVL_ST(NUMDAYS) - WSF_RES_LVL_ST_END =
WSF_RES_LVL_ST_END_ABOVE - WSF_RES_LVL_ST_END_BELOW;

[WSF_RES_CMT_ST_START] WSF_RES_CMT_ST(1) = WSF_RES_CMT_ST_INT;
[WSF_RES_CMT_ST_START2] WSF_RES_CMT_ST(2) = WSF_RES_CMT_ST_INT;
[WSF_RES_CMT_ST_END1] WSF_RES_CMT_ST(NUMDAYS) - WSF_RES_CMT_ST_END =
WSF_RES_CMT_ST_END_ABOVE - WSF_RES_CMT_ST_END_BELOW;

[WSF_RES_BBK_ST_START] WSF_RES_BBK_ST(1) = WSF_RES_BBK_ST_INT;
[WSF_RES_BBK_ST_START2] WSF_RES_BBK_ST(2) = WSF_RES_BBK_ST_INT;
[WSF_RES_BBK_ST_END1] WSF_RES_BBK_ST(NUMDAYS) - WSF_RES_BBK_ST_END =
WSF_RES_BBK_ST_END_ABOVE - WSF_RES_BBK_ST_END_BELOW;

!Farmington;
[FAR_RES_OTI_ST_START] FAR_RES_OTI_ST(1) = FAR_RES_OTI_ST_INT;
![FAR_RES_OTI_ST_START2] FAR_RES_OTI_ST(2) = FAR_RES_OTI_ST_INT;
[FAR_RES_OTI_ST_END1] FAR_RES_OTI_ST(NUMDAYS) - FAR_RES_OTI_ST_END =
FAR_RES_OTI_ST_END_ABOVE - FAR_RES_OTI_ST_END_BELOW;

[FAR_RES_COL_ST_START] FAR_RES_COL_ST(1) = FAR_RES_COL_ST_INT;
![FAR_RES_COL_ST_START2] FAR_RES_COL_ST(2) = FAR_RES_COL_ST_INT;
[FAR_RES_COL_ST_END1] FAR_RES_COL_ST(NUMDAYS) - FAR_RES_COL_ST_END =
FAR_RES_COL_ST_END_ABOVE - FAR_RES_COL_ST_END_BELOW;

[FAR_RES_WBR_ST_START] FAR_RES_WBR_ST(1) = FAR_RES_WBR_ST_INT;
![FAR_RES_WBR_ST_START2] FAR_RES_WBR_ST(2) = FAR_RES_WBR_ST_INT;
[FAR_RES_WBR_ST_END1] FAR_RES_WBR_ST(NUMDAYS) - FAR_RES_WBR_ST_END =
FAR_RES_WBR_ST_END_ABOVE - FAR_RES_WBR_ST_END_BELOW;

[FAR_RES_NEP_ST_START] FAR_RES_NEP_ST(1) = FAR_RES_NEP_ST_INT;
![FAR_RES_NEP_ST_START2] FAR_RES_NEP_ST(2) = FAR_RES_NEP_ST_INT;
[FAR_RES_NEP_ST_END1] FAR_RES_NEP_ST(NUMDAYS) - FAR_RES_NEP_ST_END =
FAR_RES_NEP_ST_END_ABOVE - FAR_RES_NEP_ST_END_BELOW;

[FAR_RES_LMD_ST_START] FAR_RES_LMD_ST(1) = FAR_RES_LMD_ST_INT;
![FAR_RES_LMD_ST_START2] FAR_RES_LMD_ST(2) = FAR_RES_LMD_ST_INT;
[FAR_RES_LMD_ST_END1] FAR_RES_LMD_ST(NUMDAYS) - FAR_RES_LMD_ST_END =
FAR_RES_LMD_ST_END_ABOVE - FAR_RES_LMD_ST_END_BELOW;

[FAR_RES_BKH_ST_START] FAR_RES_BKH_ST(1) = FAR_RES_BKH_ST_INT;
![FAR_RES_BKH_ST_START2] FAR_RES_BKH_ST(2) = FAR_RES_BKH_ST_INT;
[FAR_RES_BKH_ST_END1] FAR_RES_BKH_ST(NUMDAYS) - FAR_RES_BKH_ST_END =
FAR_RES_BKH_ST_END_ABOVE - FAR_RES_BKH_ST_END_BELOW;

[FAR_RES_RBW_ST_START] FAR_RES_RBW_ST(1) = FAR_RES_RBW_ST_INT;
![FAR_RES_RBW_ST_START2] FAR_RES_RBW_ST(2) = FAR_RES_RBW_ST_INT;
[FAR_RES_RBW_ST_END1] FAR_RES_RBW_ST(NUMDAYS) - FAR_RES_RBW_ST_END =
FAR_RES_RBW_ST_END_ABOVE - FAR_RES_RBW_ST_END_BELOW;

!Lower Third Mainstem;
[MAIN_RES_VRN_ST_START] MAIN_RES_VRN_ST(1) = MAIN_RES_VRN_ST_INT;
[MAIN_RES_VRN_ST_START2] MAIN_RES_VRN_ST(2) = MAIN_RES_VRN_ST_INT;
[MAIN_RES_VRN_ST_END1] MAIN_RES_VRN_ST(NUMDAYS) - MAIN_RES_VRN_ST_END =
MAIN_RES_VRN_ST_END_ABOVE - MAIN_RES_VRN_ST_END_BELOW;

[MAIN_RES_TRN_ST_START] MAIN_RES_TRN_ST(1) = MAIN_RES_TRN_ST_INT;
[MAIN_RES_TRN_ST_START2] MAIN_RES_TRN_ST(2) = MAIN_RES_TRN_ST_INT;
[MAIN_RES_TRN_ST_END1] MAIN_RES_TRN_ST(NUMDAYS) - MAIN_RES_TRN_ST_END =
MAIN_RES_TRN_ST_END_ABOVE - MAIN_RES_TRN_ST_END_BELOW;

[MAIN_RES_HOL_ST_START] MAIN_RES_HOL_ST(1) = MAIN_RES_HOL_ST_INT;
[MAIN_RES_HOL_ST_START2] MAIN_RES_HOL_ST(2) = MAIN_RES_HOL_ST_INT;
```

```
[MAIN_RES_HOL_ST_END1] MAIN_RES_HOL_ST(NUMDAYS) - MAIN_RES_HOL_ST_END =
MAIN_RES_HOL_ST_END_ABOVE - MAIN_RES_HOL_ST_END_BELOW;
```

Hydropower Calculations

Hydropower income is calculated for each watershed as the summation of the dams which contribute to hydropower generation. Each dam's daily income is calculated by using a location specific estimation of the relationship between flow rate, turbine power production, and historic daily energy prices.

```
!---Power Generation and Income---;

!Total Income From All Hydropower Dams By Basin;
@FOR(day(I):
[UP3_TOTAL_INCOME_L] UP3_TOTAL_INCOME(I) = MAIN_RES_MOR_INC(I) + MAIN_RES_COM_INC(I) +
MAIN_RES_MCD_INC(I);
[MID3_TOTAL_INCOME_L] MID3_TOTAL_INCOME(I) = MAIN_RES_BFA_INC(I) +
MAIN_RES_WLD_INC(I);
[LOW3_TOTAL_INCOME_L] LOW3_TOTAL_INCOME(I) = MAIN_RES_VRN_INC(I) + MAIN_RES_TRN_INC(I)
+ MAIN_RES_HOL_INC(I);
[DRF_TOTAL_INCOME_L] DRF_TOTAL_INCOME(I) = DRF_RES_SBG_INC(I) + DRF_RES_HAR_INC(I) +
DRF_RES_SHR_INC(I) + DRF_RES_DV5_INC(I) + DRF_RES_FBR_INC(I) +
                                          DRF_RES_DV4_INC(I) + DRF_RES_DV3_INC(I) +
DRF_RES_GRD_INC(I) + DRF_RES_DV2_INC(I);
[CHP_TOTAL_INCOME_L] CHP_TOTAL_INCOME(I) = CHP_RES_RBD_INC(I);
[WSF_TOTAL_INCOME_L] WSF_TOTAL_INCOME(I) = WSF_RES_CMT_INC(I);
[FAR_TOTAL_INCOME_L] FAR_TOTAL_INCOME(I) = FAR_RES_RBW_INC(I) + FAR_RES_COL_INC(I) +
FAR_RES_WBR_INC(I);
);

r = 62.428;!Specific weight of water lb/ft^3;


!Define Turbine Releases;
!Upper Third;
@FOR (day(I):
[MAIN_RES_MOR_PR_DEF] MAIN_RES_MOR_R(I) = MAIN_RES_MOR_DR(I) + MAIN_RES_MOR_PR(I);
[MAIN_RES_COM_PR_DEF] MAIN_RES_COM_R(I) = MAIN_RES_COM_DR(I) + MAIN_RES_COM_PR(I);
[MAIN_RES_MCD_PR_DEF] MAIN_RES_MCD_R(I) = MAIN_RES_MCD_DR(I) + MAIN_RES_MCD_PR(I);

!Middle Third;
[MAIN_RES_BFA_PR_DEF] MAIN_RES_BFA_R(I) = MAIN_RES_BFA_DR(I) + MAIN_RES_BFA_PR(I);
[MAIN_RES_WLD_PR_DEF] MAIN_RES_WLD_R(I) = MAIN_RES_WLD_DR(I) + MAIN_RES_WLD_PR(I);

!Deerfield;
[DRF_RES_SBG_PR_DEF] DRF_RES_SBG_R(I) = DRF_RES_SBG_PR(I) + DRF_RES_SBG_DR(I);
[DRF_RES_HAR_PR_DEF] DRF_RES_HAR_R(I) = DRF_RES_HAR_PR(I) + DRF_RES_HAR_DR(I);
[DRF_RES_SHR_PR_DEF] DRF_RES_SHR_R(I) = DRF_RES_SHR_PR(I) + DRF_RES_SHR_DR(I);
[DRF_RES_DV5_PR_DEF] DRF_RES_DV5_R(I) = DRF_RES_DV5_PR(I) + DRF_RES_DV5_DR(I);
[DRF_RES_FBR_PR_DEF] DRF_RES_FBR_R(I) = DRF_RES_FBR_PR(I) + DRF_RES_FBR_DR(I);
[DRF_RES_DV4_PR_DEF] DRF_RES_DV4_R(I) = DRF_RES_DV4_PR(I) + DRF_RES_DV4_DR(I);
[DRF_RES_DV3_PR_DEF] DRF_RES_DV3_R(I) = DRF_RES_DV3_PR(I) + DRF_RES_DV3_DR(I);
[DRF_RES_GRD_PR_DEF] DRF_RES_GRD_R(I) = DRF_RES_GRD_PR(I) + DRF_RES_GRD_DR(I);
[DRF_RES_DV2_PR_DEF] DRF_RES_DV2_R(I) = DRF_RES_DV2_PR(I) + DRF_RES_DV2_DR(I);


!Chicopee;
[CHP_RES_RBD_PR_DEF] CHP_RES_RBD_R(I) = CHP_RES_RBD_PR(I) + CHP_RES_RBD_DR(I);

!Westfield;
```

```
[WSF_RES_CMT_PR_DEF] WSF_RES_CMT_R(I) = WSF_RES_CMT_DR(I) + WSF_RES_CMT_PR(I);

!Farmington;
[FAR_RES_WBR_PR_DEF] FAR_RES_WBR_R(I) = FAR_RES_WBR_DR(I) + FAR_RES_WBR_PR(I);
[FAR_RES_COL_PR_DEF] FAR_RES_COL_R(I) = FAR_RES_COL_DR(I) + FAR_RES_COL_PR(I);
[FAR_RES_RBW_PR_DEF] FAR_RES_RBW_R(I) = FAR_RES_RBW_DR(I) + FAR_RES_RBW_PR(I);

!Lower Third Mainstem;
[MAIN_RES_VRN_PR_DEF] MAIN_RES_VRN_R(I) = MAIN_RES_VRN_DR(I) + MAIN_RES_VRN_PR(I);
[MAIN_RES_TRN_PR_DEF] MAIN_RES_TRN_R(I) = MAIN_RES_TRN_DR(I) + MAIN_RES_TRN_PR(I);
[MAIN_RES_HOL_PR_DEF] MAIN_RES_HOL_R(I) = MAIN_RES_HOL_DR(I) +
MAIN_RES_HOL_PR_CANAL(I) + MAIN_RES_HOL_PR_DAM(I);
);

!Constrain the the flow through the turbines;
!Upper Third;
@FOR (day(I):
[MAIN_RES_MOR_PR_BOUND] @BND(0,MAIN_RES_MOR_PR(I),18000*3600*24); !18,000 cfs maximum
turbine capacity;
[MAIN_RES_COM_PR_BOUND] @BND(0,MAIN_RES_COM_PR(I),11750*3600*24); !11,750 cfs maximum
turbine capacity;
[MAIN_RES_MCD_PR_BOUND] @BND(0,MAIN_RES_MCD_PR(I),6180*3600*24);  !6,180 cfs maximum
turbine capacity;

!Middle Third;
[MAIN_RES_WLD_PR_BOUND] @BND(0,MAIN_RES_WLD_PR(I),10500*3600*24); !10,500 cfs maximum
turbine capacity;
[MAIN_RES_BFA_PR_BOUND] @BND(0,MAIN_RES_BFA_PR(I),10700*3600*24); !10,700 cfs maximum
turbine capacity;

!Deerfield;
[DRF_RES_SBG_PR_BOUND] @BND(0,DRF_RES_SBG_PR(I),345*3600*24);  !345 cfs maximum
turbine capacity;
[DRF_RES_HAR_PR_BOUND] @BND(0,DRF_RES_HAR_PR(I),1800*3600*24); !1,800 cfs maximum
turbine capacity;
[DRF_RES_SHR_PR_BOUND] @BND(0,DRF_RES_SHR_PR(I),1150*3600*24); !1,150 cfs maximum
turbine capacity;
[DRF_RES_DV5_PR_BOUND] @BND(0,DRF_RES_DV5_PR(I),1000*3600*24); !1,000 cfs maximum
turbine capacity;
[DRF_RES_FBR_PR_BOUND] @BND(0,DRF_RES_FBR_PR(I),1000*3600*24); !1,000 cfs maximum
turbine capacity;
[DRF_RES_DV4_PR_BOUND] @BND(0,DRF_RES_DV4_PR(I),1500*3600*24); !1,500 cfs maximum
turbine capacity;
[DRF_RES_DV3_PR_BOUND] @BND(0,DRF_RES_DV3_PR(I),1500*3600*24); !1,500 cfs maximum
turbine capacity;
[DRF_RES_DV2_PR_BOUND] @BND(0,DRF_RES_DV2_PR(I),1500*3600*24); !1,500 cfs maximum
turbine capacity;

!Westfield;
[WSF_RES_CMT_PR_BOUND] @BND(0,WSF_RES_CMT_PR(I),(928*3600*24-WSF_RES_CMT_W(I))); !928
cfs (600 MGD) maximum power tunnel capacity;

!Farmington;
[FAR_RES_COL_PR_BOUND] @BND(0,FAR_RES_COL_PR(I),650*3600*24); !650 cfs maximum turbine
capacity;
[FAR_RES_RBW_PR_BOUND] @BND(0,FAR_RES_RBW_PR(I),2400*3600*24); !2400 cfs maximum
turbine capacity;

!Lower Third Mainstem;
[MAIN_RES_VRN_PR_BOUND] @BND(0,MAIN_RES_VRN_PR(I),15000*3600*24); !15,000 cfs maximum
turbine capacity;
[MAIN_RES_TRN_PR_BOUND] @BND(0,MAIN_RES_TRN_PR(I),16000*3600*24); !16,000 cfs is
design flow of the canal;
```

```
[MAIN_RES_HOL_PR_CANAL_BOUND] @BND(0,MAIN_RES_HOL_PR_CANAL(I),6600*3600*24); !6,600
cfs maximum canal capacity;
);



!Constrain the power generated;
@FOR (day(I):
!Upper Third;
[MAIN_RES_MOR_P_BOUND] @BND(0,MAIN_RES_MOR_P(I),192*24); !192 MW capacity, we multiply
by 24 to get bound in MWHs;
[MAIN_RES_COM_P_BOUND] @BND(0,MAIN_RES_COM_P(I),145*24); !145 MW capacity, we multiply
by 24 to get bound in MWHs;
[MAIN_RES_MCD_P_BOUND] @BND(0,MAIN_RES_MCD_P(I),10.6*24); !10.3 MW capacity, we
multiply by 24 to get bound in MWHs;
!Middle Third;
[MAIN_RES_WLD_P_BOUND] @BND(0,MAIN_RES_WLD_P(I),42*24); !42 MW capacity, we multiply
by 24 to get bound in MWHs;
[MAIN_RES_BFA_P_BOUND] @BND(0,MAIN_RES_BFA_P(I),45*24); !45 MW capacity, we multiply
by 24 to get bound in MWHs;
!Deerfield;
[DRF_RES_SBG_P_BOUND] @BND(0,DRF_RES_SBG_P(I),5*24); !5 MW capacity, we multiply by 24
to get bound in MWHs;
[DRF_RES_HAR_P_BOUND] @BND(0,DRF_RES_HAR_P(I),45*24); !45 MW capacity, we multiply by
24 to get bound in MWHs;
[DRF_RES_SHR_P_BOUND] @BND(0,DRF_RES_SHR_P(I),6.4*24); !6.4 MW capacity, we multiply
by 24 to get bound in MWHs;
[DRF_RES_DV5_P_BOUND] @BND(0,DRF_RES_DV5_P(I),12*24); !12 MW capacity, we multiply by
24 to get bound in MWHs;
[DRF_RES_FBR_P_BOUND] @BND(0,DRF_RES_FBR_P(I),9*24); !9 MW capacity, we multiply by 24
to get bound in MWHs;
[DRF_RES_DV4_P_BOUND] @BND(0,DRF_RES_DV4_P(I),6*24); !6 MW capacity, we multiply by 24
to get bound in MWHs;
[DRF_RES_DV3_P_BOUND] @BND(0,DRF_RES_DV3_P(I),6*24); !6 MW capacity, we multiply by 24
to get bound in MWHs;
[DRF_RES_GRD_P_BOUND] @BND(0,DRF_RES_GRD_P(I),3.58*24); !3.58 MW capacity, we multiply
by 24 to get bound in MWHs;
[DRF_RES_DV2_P_BOUND] @BND(0,DRF_RES_DV2_P(I),6*24); !6 MW capacity, we multiply by 24
to get bound in MWHs;
!Westfield;
[WSF_RES_CMT_P_BOUND] @BND(0,WSF_RES_CMT_P(I),33*24); !33 MW capacity, we multiply by
24 to get bound in MWHs;
!Chicopee;
[CHP_RES_RBD_P_BOUND] @BND(0,CHP_RES_RBD_P(I),3.6*24); !3.6 MW capacity, we multiply
by 24 to get bound in MWHs;
!Farmington;
[FAR_RES_RBW_P_BOUND] @BND(0,FAR_RES_RBW_P(I),10*24); !10 MW capacity, we multiply by
24 to get bound in MWHs;
[FAR_RES_COL_P_BOUND] @BND(0,FAR_RES_COL_P(I),4.5*24); !4.5 MW capacity, we multiply
by 24 to get bound in MWHs;
[FAR_RES_WBR_P_BOUND] @BND(0,FAR_RES_WBR_P(I),3.39*24); !3.39 MW capacity, we multiply
by 24 to get bound in MWHs;
!Lower Third;
[MAIN_RES_VRN_P_BOUND] @BND(0,MAIN_RES_VRN_P(I),36.8*24); !36.8 MW capacity, we
multiply by 24 to get bound in MWHs;
[MAIN_RES_TRN_P_BOUND] @BND(0,MAIN_RES_TRN_P(I),67.71*24); !67.71 MW FERC license
allowance, we multiply by 24 to get bound in MWHs;
[MAIN_RES_HOL_P_CANAL_BOUND] @BND(0,MAIN_RES_HOL_P_CANAL(I),13.8*24); !13.8 MW
capacity, we multiply by 24 to get bound in MWHs;
[MAIN_RES_HOL_P_DAM_BOUND] @BND(0,MAIN_RES_HOL_P_DAM(I),30*24); !30 MW capacity, we
multiply by 24 to get bound in MWHs;
);
```

```
!Calculate the power production;
!Power calculated as power =r [lb/ft^3] *  head [ft] * n [%]  * flow [ft^3/day] =
[ft*lb/day]. Then multiplied by 24 and 6.37253568*10^10 to get MWH produced that day;
!For Transcanada Dams, we were provided with power produced (in MW) from flow rates.
We multiply by 24 to get the MWH produced that day;
!Efficiencies: Back-calculate from maximum power production under max net head and
maximum flow.
!              If no info is available, then 0.6 for small hydro and 0.9 for larger
hydro;

@FOR (day(I):
!Upper Third;
[MAIN_RES_MOR_P_L] MAIN_RES_MOR_P(I) = 24*(MAIN_RES_MOR_PR(I)/86400)/92; !MWH per day
produced ==> 92 cfs per MW produced;
[MAIN_RES_COM_P_L] MAIN_RES_COM_P(I) = 24*(MAIN_RES_COM_PR(I)/86400)/77; !MWH per day
produced ==> 77 cfs per MW produced;
[MAIN_RES_MCD_P_L] MAIN_RES_MCD_P(I) = 24*(MAIN_RES_MCD_PR(I)/86400)/600; !MWH per day
produced ==> 600 cfs per MW produced;

!Middle Third;
[MAIN_RES_WLD_P_L] MAIN_RES_WLD_P(I) = 24*(MAIN_RES_WLD_PR(I)/86400)/265;  !MWH per
day produced ==> 265 cfs per MW produced;
[MAIN_RES_BFA_P_L] MAIN_RES_BFA_P(I) = 24*(MAIN_RES_BFA_PR(I)/86400)/218;  !MWH per
day produced ==> 218 cfs per MW produced;


!Deefield;
[DRF_RES_SBG_P_L] DRF_RES_SBG_P(I) = 24*(DRF_RES_SBG_PR(I)/86400)/69;  !MWH per day
produced ==> 69 cfs per MW produced;
[DRF_RES_HAR_P_L] DRF_RES_HAR_P(I) = 24*(DRF_RES_HAR_PR(I)/86400)/42;  !MWH per day
produced ==> 42 cfs per MW produced;
[DRF_RES_SHR_P_L] DRF_RES_SHR_P(I) = 24*(DRF_RES_SHR_PR(I)/86400)/179;  !MWH per day
produced ==> 179 cfs per MW produced;
[DRF_RES_DV5_P_L] DRF_RES_DV5_P(I) = 24*(DRF_RES_DV5_PR(I)/86400)/95;  !MWH per day
produced ==> 95 cfs per MW produced;
[DRF_RES_FBR_P_L] DRF_RES_FBR_P(I) = 24*(DRF_RES_FBR_PR(I)/86400)/150;  !MWH per day
produced ==> 150 cfs per MW produced;
[DRF_RES_DV4_P_L] DRF_RES_DV4_P(I) = 24*(DRF_RES_DV4_PR(I)/86400)/251;  !MWH per day
produced ==> 251 cfs per MW produced;
[DRF_RES_DV3_P_L] DRF_RES_DV3_P(I) = 24*(DRF_RES_DV3_PR(I)/86400)/213;  !MWH per day
produced ==> 213 cfs per MW produced;
[DRF_RES_GRD_P_L] DRF_RES_GRD_P(I) = r * 30 * 0.6 *
DRF_RES_GRD_PR(I)*24*(6.37253568*(10^10)); !MWH per day produced ==> 30 ft max dam
height, efficiency of 0.6;
[DRF_RES_DV2_P_L] DRF_RES_DV2_P(I) = 24*(DRF_RES_DV2_PR(I)/86400)/238;  !MWH per day
produced ==> 238 cfs per MW produced;

!Chicopee;
[CHP_RES_RBD_P_L] CHP_RES_RBD_P(I) = r * 51 * 0.6 *
CHP_RES_RBD_PR(I)*24*(6.37253568*(10^10)); !MWH per day produced ==> 51 ft max dam
height, efficiency of 0.6;

!Westfield;
[WSF_RES_CMT_P_L] WSF_RES_CMT_P(I) = (7.48/10^6)*(WSF_RES_CMT_PR(I)+WSF_RES_CMT_W(I));
!MWH per day produced ==> for CMT, 1MG passed is equivalent to 1 MWH produced ==> we
multiply by (7.48/10^6) to get MGD released;

!Farmington;
[FAR_RES_RBW_P_L] FAR_RES_RBW_P(I) = r * 62 * 0.6 *
FAR_RES_RBW_PR(I)*24*(6.37253568*(10^10)); !MWH per day produced ==> 62 ft max dam
height, efficiency of 0.6;
```

```
[FAR_RES_COL_P_L] FAR_RES_COL_P(I) = r * 110 * 0.9 *
FAR_RES_COL_PR(I)*24*(6.37253568*(10^10)); !MWH per day produced ==> 110 ft max dam
height, efficiency of 0.9;
[FAR_RES_WBR_P_L] FAR_RES_WBR_P(I) = r * 135 * 0.9 *
FAR_RES_WBR_PR(I)*24*(6.37253568*(10^10)); !MWH per day produced ==> 135 ft max dam
height, efficiency of 0.9;

!Lower Third Mainstem;
[MAIN_RES_VRN_P_L] MAIN_RES_VRN_P(I) = 24*(MAIN_RES_VRN_PR(I)/86400)/453;  !MWH per
day produced ==> 453 cfs per MW produced;
[MAIN_RES_TRN_P_L] MAIN_RES_TRN_P(I) = r * 50 * 0.9 *
MAIN_RES_TRN_PR(I)*24*(6.37253568*(10^10)); !MWH per day produced ==> 50 ft net max
head, efficiency of 0.9;
[MAIN_RES_HOL_P_CANAL_L] MAIN_RES_HOL_P_CANAL(I) = r * 30 * 0.9 *
MAIN_RES_HOL_PR_CANAL(I)*24*(6.37253568*(10^10)); !MWH per day produced ==> 30 ft net
max head, efficiency of 0.9;
[MAIN_RES_HOL_P_DAM_L] MAIN_RES_HOL_P_DAM(I) = r * 30 * 0.9 *
MAIN_RES_HOL_PR_DAM(I)*24*(6.37253568*(10^10)); !MWH per day produced ==> 30 ft net
max head, efficiency of 0.9;

);


!Calculate the income;
@FOR (day(I):
!Upper Third;
[MAIN_RES_MOR_INC_L] MAIN_RES_MOR_INC(I) = MAIN_RES_MOR_P(I)*ENERGY_PRICE(I);
[MAIN_RES_COM_INC_L] MAIN_RES_COM_INC(I) = MAIN_RES_COM_P(I)*ENERGY_PRICE(I);
[MAIN_RES_MCD_INC_L] MAIN_RES_MCD_INC(I) = MAIN_RES_MCD_P(I)*ENERGY_PRICE(I);

!Middle Third;
[MAIN_RES_BFA_INC_L] MAIN_RES_BFA_INC(I) = MAIN_RES_BFA_P(I)*ENERGY_PRICE(I);
[MAIN_RES_WLD_INC_L] MAIN_RES_WLD_INC(I) = MAIN_RES_WLD_P(I)*ENERGY_PRICE(I);

!Deerfield;
[DRF_RES_SBG_INC_L] DRF_RES_SBG_INC(I) = DRF_RES_SBG_P(I)*ENERGY_PRICE(I);
[DRF_RES_HAR_INC_L] DRF_RES_HAR_INC(I) = DRF_RES_HAR_P(I)*ENERGY_PRICE(I);
[DRF_RES_SHR_INC_L] DRF_RES_SHR_INC(I) = DRF_RES_SHR_P(I)*ENERGY_PRICE(I);
[DRF_RES_DV5_INC_L] DRF_RES_DV5_INC(I) = DRF_RES_DV5_P(I)*ENERGY_PRICE(I);
[DRF_RES_FBR_INC_L] DRF_RES_FBR_INC(I) = DRF_RES_FBR_P(I)*ENERGY_PRICE(I);
[DRF_RES_DV4_INC_L] DRF_RES_DV4_INC(I) = DRF_RES_DV4_P(I)*ENERGY_PRICE(I);
[DRF_RES_DV3_INC_L] DRF_RES_DV3_INC(I) = DRF_RES_DV3_P(I)*ENERGY_PRICE(I);
[DRF_RES_GRD_INC_L] DRF_RES_GRD_INC(I) = DRF_RES_GRD_P(I)*ENERGY_PRICE(I);
[DRF_RES_DV2_INC_L] DRF_RES_DV2_INC(I) = DRF_RES_DV2_P(I)*ENERGY_PRICE(I);

!Chicopee;
[CHP_RES_RBD_INC_L] CHP_RES_RBD_INC(I) = CHP_RES_RBD_P(I)*ENERGY_PRICE(I);

!Westfield;
[WSF_RES_CMT_INC_L] WSF_RES_CMT_INC(I) = WSF_RES_CMT_P(I)*ENERGY_PRICE(I);

!Farmington;
[FAR_RES_RBW_INC_L] FAR_RES_RBW_INC(I) = FAR_RES_RBW_P(I)*ENERGY_PRICE(I);
[FAR_RES_COL_INC_L] FAR_RES_COL_INC(I) = FAR_RES_COL_P(I)*ENERGY_PRICE(I);
[FAR_RES_WBR_INC_L] FAR_RES_WBR_INC(I) = FAR_RES_WBR_P(I)*ENERGY_PRICE(I);

!Lower Third Mainstem;
[MAIN_RES_VRN_INC_L] MAIN_RES_VRN_INC(I) = MAIN_RES_VRN_P(I)*ENERGY_PRICE(I);
[MAIN_RES_TRN_INC_L] MAIN_RES_TRN_INC(I) = MAIN_RES_TRN_P(I)*ENERGY_PRICE(I);
[MAIN_RES_HOL_INC_L] MAIN_RES_HOL_INC(I) = (MAIN_RES_HOL_P_CANAL(I) +
MAIN_RES_HOL_P_DAM(I))* ENERGY_PRICE(I);
);
```

Econode Calculations

Similar to the Variable Definitions section, this section mathematically defines necessary variables for the objective function. Specifically, the modeled flow rates for each eco-node are defined as the summation of modeled reservoir releases and side-flows which represent contributing flows downstream of these modeled releases.

```
!ECONODE MASS BALANCE;

!Time lags are ignored in the first time step;
@FOR (day(I) | I #LE# 1:
[MAIN_EN_1_MB1] MAIN_EN_1_MOD(I) = MAIN_EN_1_Q(I) + UCN_RES_LFR_R(I);
[MAIN_EN_6_MB1] MAIN_EN_6_MOD(I) = MAIN_EN_6_Q(I) + MAIN_RES_COM_R(I);
[MAIN_EN_9_MB1] MAIN_EN_9_MOD(I) = MAIN_EN_9_Q(I) + OMP_RES_UNV_R(I) +
MAIN_RES_MCD_R(I);
[MAIN_EN_11_MB1] MAIN_EN_11_MOD(I) = MAIN_EN_11_Q(I) + MAIN_RES_WLD_R(I) +
MAS_RES_MSL_R(I) + OTT_RES_NHD_R(I);
[MAS_EN_19_MB1] MAS_EN_19_MOD(I) = MAS_EN_19_Q(I) + MAS_RES_GOO_R(I) +
MAS_RES_CRY_R(I);
[BLK_EN_21_MB1] BLK_EN_21_MOD(I) = BLK_EN_21_Q(I) + BLK_RES_NSP_R(I);
[SGR_EN_22_MB1] SGR_EN_22_MOD(I) = SGR_EN_22_Q(I) + SUG_RES_SGR_R(I);
[MAIN_EN_24_MB1] MAIN_EN_24_MOD(I) = MAIN_EN_24_Q(I) + MAIN_RES_BFA_R(I);
[MAIN_EN_26_MB1] MAIN_EN_26_MOD(I) = MAIN_EN_26_Q(I) + MAIN_RES_VRN_R(I) +
ASH_RES_SMD_R(I) + ASH_RES_OBD_R(I);
[WST_EN_33_MB1] WST_EN_33_MOD(I) = WST_EN_33_Q(I) + WST_RES_TWN_R(I);
[ASH_EN_37_MB1] ASH_EN_37_MOD(I) = ASH_EN_37_Q(I) + ASH_RES_SMD_R(I) +
ASH_RES_OBD_R(I);
[MLR_EN_39_MB1] MLR_EN_39_MOD(I) = MLR_EN_39_Q(I) + MLR_RES_TUL_R(I) +
MLR_RES_BIR_R(I);
[DRF_EN_45_MB1] DRF_EN_45_MOD(I) = DRF_EN_45_Q(I) + DRF_RES_DV2_R(I);
[FAR_EN_59_MB1] FAR_EN_59_MOD(I) = FAR_EN_59_Q(I) + FAR_RES_WBR_R(I) +
FAR_RES_LMD_R(I) + FAR_RES_NEP_R(I);
[MAIN_EN_74_MB1] MAIN_EN_74_MOD(I) = MAIN_EN_74_Q(I) + WSF_RES_KVL_R(I) +
WSF_RES_LVL_R(I) + WSF_RES_CMT_R(I) + MAIN_RES_HOL_R(I) + CHP_RES_RBD_R(I);
[WSF_EN_76_MB1] WSF_EN_76_MOD(I) = WSF_EN_76_Q(I) + WSF_RES_KVL_R(I) +
WSF_RES_LVL_R(I) + WSF_RES_CMT_R(I);
[WSF_EN_78_MB1] WSF_EN_78_MOD(I) = WSF_EN_78_Q(I) + WSF_RES_KVL_R(I) +
WSF_RES_LVL_R(I);
[CHP_EN_80_MB1] CHP_EN_80_MOD(I) = CHP_EN_80_Q(I) + CHP_RES_BFD_R(I);
[MAIN_EN_88_MB1] MAIN_EN_88_MOD(I) = MAIN_EN_88_Q(I) + DRF_RES_DV2_R(I) +
MAIN_RES_TRN_R(I);
[ASH_EN_91_MB1] ASH_EN_91_MOD(I) = ASH_EN_91_Q(I) + ASH_RES_SMD_R(I);
[WST_EN_95_MB1] WST_EN_95_MOD(I) = WST_EN_95_Q(I) + WST_RES_BMD_R(I);
[MAIN_EN_98_MB1] MAIN_EN_98_MOD(I) = MAIN_EN_98_Q(I) + MAIN_RES_WLD_R(I) +
MAS_RES_MSL_R(I);
[MAIN_EN_100_MB1] MAIN_EN_100_MOD(I) = MAIN_EN_100_Q(I) + MAIN_RES_MCD_R(I);
[DRF_EN_136_MB1] DRF_EN_136_MOD(I) = DRF_EN_136_Q(I) + DRF_RES_HAR_R(I);
[DRF_EN_137_MB1] DRF_EN_137_MOD(I) = DRF_EN_137_Q(I) + DRF_RES_FBR_R(I);
[FAR_EN_139_MB1] FAR_EN_139_MOD(I) = FAR_EN_139_Q(I) + FAR_RES_BKH_R(I);
[FAR_EN_149_MB1] FAR_EN_149_MOD(I) = FAR_EN_149_Q(I) + FAR_RES_RBW_R(I);
[MAIN_EN_152_MB1] MAIN_EN_152_MOD(I) = MAIN_EN_152_Q(I) + MAIN_RES_HOL_R(I);
);

@FOR (day(I) | I #GE# 2:
[MAIN_EN_1_MB2] MAIN_EN_1_MOD(I) = MAIN_EN_1_Q(I) + UCN_RES_LFR_R(I);
[MAIN_EN_6_MB2] MAIN_EN_6_MOD(I) = MAIN_EN_6_Q(I) + MAIN_RES_COM_R(I);
[MAIN_EN_9_MB2] MAIN_EN_9_MOD(I) = MAIN_EN_9_Q(I) + OMP_RES_UNV_R(I) +
MAIN_RES_MCD_R(I-1);
```

```
[MAIN_EN_11_MB2] MAIN_EN_11_MOD(I) = MAIN_EN_11_Q(I) + MAIN_RES_WLD_R(I) +
MAS_RES_MSL_R(I) + OTT_RES_NHD_R(I);
[MAS_EN_19_MB2] MAS_EN_19_MOD(I) = MAS_EN_19_Q(I) + MAS_RES_GOO_R(I) +
MAS_RES_CRY_R(I);
[BLK_EN_21_MB2] BLK_EN_21_MOD(I) = BLK_EN_21_Q(I) + BLK_RES_NSP_R(I);
[SGR_EN_22_MB2] SGR_EN_22_MOD(I) = SGR_EN_22_Q(I) + SUG_RES_SGR_R(I);
[MAIN_EN_24_MB2] MAIN_EN_24_MOD(I) = MAIN_EN_24_Q(I) + MAIN_RES_BFA_R(I);
[MAIN_EN_26_MB2] MAIN_EN_26_MOD(I) = MAIN_EN_26_Q(I) + MAIN_RES_VRN_R(I) +
ASH_RES_SMD_R(I) + ASH_RES_OBD_R(I);
[WST_EN_33_MB2] WST_EN_33_MOD(I) = WST_EN_33_Q(I) + WST_RES_TWN_R(I);
[ASH_EN_37_MB2] ASH_EN_37_MOD(I) = ASH_EN_37_Q(I) + ASH_RES_SMD_R(I) +
ASH_RES_OBD_R(I);
[MLR_EN_39_MB2] MLR_EN_39_MOD(I) = MLR_EN_39_Q(I) + MLR_RES_TUL_R(I) +
MLR_RES_BIR_R(I);
[DRF_EN_45_MB2] DRF_EN_45_MOD(I) = DRF_EN_45_Q(I) + DRF_RES_DV2_R(I);
[FAR_EN_59_MB2] FAR_EN_59_MOD(I) = FAR_EN_59_Q(I) + FAR_RES_WBR_R(I) +
FAR_RES_LMD_R(I) + FAR_RES_NEP_R(I);
[MAIN_EN_74_MB2] MAIN_EN_74_MOD(I) = MAIN_EN_74_Q(I) + WSF_RES_KVL_R(I) +
WSF_RES_LVL_R(I) + WSF_RES_CMT_R(I) + MAIN_RES_HOL_R(I) + CHP_RES_RBD_R(I);
[WSF_EN_76_MB2] WSF_EN_76_MOD(I) = WSF_EN_76_Q(I) + WSF_RES_KVL_R(I) +
WSF_RES_LVL_R(I) + WSF_RES_CMT_R(I);
[WSF_EN_78_MB2] WSF_EN_78_MOD(I) = WSF_EN_78_Q(I) + WSF_RES_KVL_R(I) +
WSF_RES_LVL_R(I);
[CHP_EN_80_MB2] CHP_EN_80_MOD(I) = CHP_EN_80_Q(I) + CHP_RES_BFD_R(I);
[MAIN_EN_88_MB2] MAIN_EN_88_MOD(I) = MAIN_EN_88_Q(I) + DRF_RES_DV2_R(I-1) +
MAIN_RES_TRN_R(I-1);
[ASH_EN_91_MB2] ASH_EN_91_MOD(I) = ASH_EN_91_Q(I) + ASH_RES_SMD_R(I);
[WST_EN_95_MB2] WST_EN_95_MOD(I) = WST_EN_95_Q(I) + WST_RES_BMD_R(I);
[MAIN_EN_98_MB2] MAIN_EN_98_MOD(I) = MAIN_EN_98_Q(I) + MAIN_RES_WLD_R(I) +
MAS_RES_MSL_R(I);
[MAIN_EN_100_MB2] MAIN_EN_100_MOD(I) = MAIN_EN_100_Q(I) + MAIN_RES_MCD_R(I);
[DRF_EN_136_MB2] DRF_EN_136_MOD(I) = DRF_EN_136_Q(I) + DRF_RES_HAR_R(I);
[DRF_EN_137_MB2] DRF_EN_137_MOD(I) = DRF_EN_137_Q(I) + DRF_RES_FBR_R(I);
[FAR_EN_139_MB2] FAR_EN_139_MOD(I) = FAR_EN_139_Q(I) + FAR_RES_BKH_R(I);
[FAR_EN_149_MB2] FAR_EN_149_MOD(I) = FAR_EN_149_Q(I) + FAR_RES_RBW_R(I);
[MAIN_EN_152_MB2] MAIN_EN_152_MOD(I) = MAIN_EN_152_Q(I) + MAIN_RES_HOL_R(I);
);

!CALCULATE PERCENT DEVIATIONS;
@FOR (day(I):
[EN_1_LOW2_CALC]  MAIN_EN_1_MOD(I) - (1+MAIN_EN_1_LOW2(I))*MAIN_EN_1_NAT(I) =
MAIN_EN_1_LOW2_ABOVE(I) - MAIN_EN_1_LOW2_BELOW(I);
[EN_1_LOW_TOT_CALC]  MAIN_EN_1_MOD(I) - (1+MAIN_EN_1_LOW1(I))*MAIN_EN_1_NAT(I) =
MAIN_EN_1_LOW1_ABOVE(I) - MAIN_EN_1_TOTAL_BELOW(I);
[EN_1_LOW1_CALC]  MAIN_EN_1_LOW1_BELOW(I) = MAIN_EN_1_TOTAL_BELOW(I) -
MAIN_EN_1_LOW2_BELOW(I);

[EN_1_HIGH_TOT_CALC] MAIN_EN_1_MOD(I) - (1+MAIN_EN_1_HIGH1(I))*MAIN_EN_1_NAT(I) =
MAIN_EN_1_TOTAL_ABOVE(I) - MAIN_EN_1_HIGH1_BELOW(I);
[EN_1_HIGH2_CALC] MAIN_EN_1_MOD(I) - (1+MAIN_EN_1_HIGH2(I))*MAIN_EN_1_NAT(I) =
MAIN_EN_1_HIGH2_ABOVE(I) - MAIN_EN_1_HIGH2_BELOW(I);
[EN_1_HIGH1_CALC] MAIN_EN_1_HIGH1_ABOVE(I) = MAIN_EN_1_TOTAL_ABOVE(I) -
MAIN_EN_1_HIGH2_ABOVE(I);


[EN_6_LOW2_CALC]  MAIN_EN_6_MOD(I) - (1+MAIN_EN_6_LOW2(I))*MAIN_EN_6_NAT(I) =
MAIN_EN_6_LOW2_ABOVE(I) - MAIN_EN_6_LOW2_BELOW(I);
[EN_6_LOW_TOT_CALC]  MAIN_EN_6_MOD(I) - (1+MAIN_EN_6_LOW1(I))*MAIN_EN_6_NAT(I) =
MAIN_EN_6_LOW1_ABOVE(I) - MAIN_EN_6_TOTAL_BELOW(I);
[EN_6_LOW1_CALC]  MAIN_EN_6_LOW1_BELOW(I) = MAIN_EN_6_TOTAL_BELOW(I) -
MAIN_EN_6_LOW2_BELOW(I);
```

```
[EN_6_HIGH_TOT_CALC] MAIN_EN_6_MOD(I) - (1+MAIN_EN_6_HIGH1(I))*MAIN_EN_6_NAT(I) =
MAIN_EN_6_TOTAL_ABOVE(I) - MAIN_EN_6_HIGH1_BELOW(I);
[EN_6_HIGH2_CALC] MAIN_EN_6_MOD(I) - (1+MAIN_EN_6_HIGH2(I))*MAIN_EN_6_NAT(I) =
MAIN_EN_6_HIGH2_ABOVE(I) - MAIN_EN_6_HIGH2_BELOW(I);
[EN_6_HIGH1_CALC] MAIN_EN_6_HIGH1_ABOVE(I) = MAIN_EN_6_TOTAL_ABOVE(I) -
MAIN_EN_6_HIGH2_ABOVE(I);


[EN_9_LOW2_CALC]  MAIN_EN_9_MOD(I) - (1+MAIN_EN_9_LOW2(I))*MAIN_EN_9_NAT(I) =
MAIN_EN_9_LOW2_ABOVE(I) - MAIN_EN_9_LOW2_BELOW(I);
[EN_9_LOW_TOT_CALC]  MAIN_EN_9_MOD(I) - (1+MAIN_EN_9_LOW1(I))*MAIN_EN_9_NAT(I) =
MAIN_EN_9_LOW1_ABOVE(I) - MAIN_EN_9_TOTAL_BELOW(I);
[EN_9_LOW1_CALC]  MAIN_EN_9_LOW1_BELOW(I) = MAIN_EN_9_TOTAL_BELOW(I) -
MAIN_EN_9_LOW2_BELOW(I);

[EN_9_HIGH_TOT_CALC] MAIN_EN_9_MOD(I) - (1+MAIN_EN_9_HIGH1(I))*MAIN_EN_9_NAT(I) =
MAIN_EN_9_TOTAL_ABOVE(I) - MAIN_EN_9_HIGH1_BELOW(I);
[EN_9_HIGH2_CALC] MAIN_EN_9_MOD(I) - (1+MAIN_EN_9_HIGH2(I))*MAIN_EN_9_NAT(I) =
MAIN_EN_9_HIGH2_ABOVE(I) - MAIN_EN_9_HIGH2_BELOW(I);
[EN_9_HIGH1_CALC] MAIN_EN_9_HIGH1_ABOVE(I) = MAIN_EN_9_TOTAL_ABOVE(I) -
MAIN_EN_9_HIGH2_ABOVE(I);


[EN_11_LOW2_CALC]  MAIN_EN_11_MOD(I) - (1+MAIN_EN_11_LOW2(I))*MAIN_EN_11_NAT(I) =
MAIN_EN_11_LOW2_ABOVE(I) - MAIN_EN_11_LOW2_BELOW(I);
[EN_11_LOW_TOT_CALC]  MAIN_EN_11_MOD(I) - (1+MAIN_EN_11_LOW1(I))*MAIN_EN_11_NAT(I) =
MAIN_EN_11_LOW1_ABOVE(I) - MAIN_EN_11_TOTAL_BELOW(I);
[EN_11_LOW1_CALC]  MAIN_EN_11_LOW1_BELOW(I) = MAIN_EN_11_TOTAL_BELOW(I) -
MAIN_EN_11_LOW2_BELOW(I);

[EN_11_HIGH_TOT_CALC] MAIN_EN_11_MOD(I) - (1+MAIN_EN_11_HIGH1(I))*MAIN_EN_11_NAT(I) =
MAIN_EN_11_TOTAL_ABOVE(I) - MAIN_EN_11_HIGH1_BELOW(I);
[EN_11_HIGH2_CALC] MAIN_EN_11_MOD(I) - (1+MAIN_EN_11_HIGH2(I))*MAIN_EN_11_NAT(I) =
MAIN_EN_11_HIGH2_ABOVE(I) - MAIN_EN_11_HIGH2_BELOW(I);
[EN_11_HIGH1_CALC] MAIN_EN_11_HIGH1_ABOVE(I) = MAIN_EN_11_TOTAL_ABOVE(I) -
MAIN_EN_11_HIGH2_ABOVE(I);


[EN_19_LOW2_CALC]  MAS_EN_19_MOD(I) - (1+MAS_EN_19_LOW2(I))*MAS_EN_19_NAT(I) =
MAS_EN_19_LOW2_ABOVE(I) - MAS_EN_19_LOW2_BELOW(I);
[EN_19_LOW_TOT_CALC]  MAS_EN_19_MOD(I) - (1+MAS_EN_19_LOW1(I))*MAS_EN_19_NAT(I) =
MAS_EN_19_LOW1_ABOVE(I) - MAS_EN_19_TOTAL_BELOW(I);
[EN_19_LOW1_CALC]  MAS_EN_19_LOW1_BELOW(I) = MAS_EN_19_TOTAL_BELOW(I) -
MAS_EN_19_LOW2_BELOW(I);

[EN_19_HIGH_TOT_CALC] MAS_EN_19_MOD(I) - (1+MAS_EN_19_HIGH1(I))*MAS_EN_19_NAT(I) =
MAS_EN_19_TOTAL_ABOVE(I) - MAS_EN_19_HIGH1_BELOW(I);
[EN_19_HIGH2_CALC] MAS_EN_19_MOD(I) - (1+MAS_EN_19_HIGH2(I))*MAS_EN_19_NAT(I) =
MAS_EN_19_HIGH2_ABOVE(I) - MAS_EN_19_HIGH2_BELOW(I);
[EN_19_HIGH1_CALC] MAS_EN_19_HIGH1_ABOVE(I) = MAS_EN_19_TOTAL_ABOVE(I) -
MAS_EN_19_HIGH2_ABOVE(I);


[EN_21_LOW2_CALC]  BLK_EN_21_MOD(I) - (1+BLK_EN_21_LOW2(I))*BLK_EN_21_NAT(I) =
BLK_EN_21_LOW2_ABOVE(I) - BLK_EN_21_LOW2_BELOW(I);
[EN_21_LOW_TOT_CALC]  BLK_EN_21_MOD(I) - (1+BLK_EN_21_LOW1(I))*BLK_EN_21_NAT(I) =
BLK_EN_21_LOW1_ABOVE(I) - BLK_EN_21_TOTAL_BELOW(I);
[EN_21_LOW1_CALC]  BLK_EN_21_LOW1_BELOW(I) = BLK_EN_21_TOTAL_BELOW(I) -
BLK_EN_21_LOW2_BELOW(I);

[EN_21_HIGH_TOT_CALC] BLK_EN_21_MOD(I) - (1+BLK_EN_21_HIGH1(I))*BLK_EN_21_NAT(I) =
BLK_EN_21_TOTAL_ABOVE(I) - BLK_EN_21_HIGH1_BELOW(I);
```

```
[EN_21_HIGH2_CALC] BLK_EN_21_MOD(I) - (1+BLK_EN_21_HIGH2(I))*BLK_EN_21_NAT(I) =
BLK_EN_21_HIGH2_ABOVE(I) - BLK_EN_21_HIGH2_BELOW(I);
[EN_21_HIGH1_CALC] BLK_EN_21_HIGH1_ABOVE(I) = BLK_EN_21_TOTAL_ABOVE(I) -
BLK_EN_21_HIGH2_ABOVE(I);


[EN_22_LOW2_CALC]  SGR_EN_22_MOD(I) - (1+SGR_EN_22_LOW2(I))*SGR_EN_22_NAT(I) =
SGR_EN_22_LOW2_ABOVE(I) - SGR_EN_22_LOW2_BELOW(I);
[EN_22_LOW_TOT_CALC]  SGR_EN_22_MOD(I) - (1+SGR_EN_22_LOW1(I))*SGR_EN_22_NAT(I) =
SGR_EN_22_LOW1_ABOVE(I) - SGR_EN_22_TOTAL_BELOW(I);
[EN_22_LOW1_CALC]  SGR_EN_22_LOW1_BELOW(I) = SGR_EN_22_TOTAL_BELOW(I) -
SGR_EN_22_LOW2_BELOW(I);

[EN_22_HIGH_TOT_CALC] SGR_EN_22_MOD(I) - (1+SGR_EN_22_HIGH1(I))*SGR_EN_22_NAT(I) =
SGR_EN_22_TOTAL_ABOVE(I) - SGR_EN_22_HIGH1_BELOW(I);
[EN_22_HIGH2_CALC] SGR_EN_22_MOD(I) - (1+SGR_EN_22_HIGH2(I))*SGR_EN_22_NAT(I) =
SGR_EN_22_HIGH2_ABOVE(I) - SGR_EN_22_HIGH2_BELOW(I);
[EN_22_HIGH1_CALC] SGR_EN_22_HIGH1_ABOVE(I) = SGR_EN_22_TOTAL_ABOVE(I) -
SGR_EN_22_HIGH2_ABOVE(I);


[EN_24_LOW2_CALC]  MAIN_EN_24_MOD(I) - (1+MAIN_EN_24_LOW2(I))*MAIN_EN_24_NAT(I) =
MAIN_EN_24_LOW2_ABOVE(I) - MAIN_EN_24_LOW2_BELOW(I);
[EN_24_LOW_TOT_CALC]  MAIN_EN_24_MOD(I) - (1+MAIN_EN_24_LOW1(I))*MAIN_EN_24_NAT(I) =
MAIN_EN_24_LOW1_ABOVE(I) - MAIN_EN_24_TOTAL_BELOW(I);
[EN_24_LOW1_CALC]  MAIN_EN_24_LOW1_BELOW(I) = MAIN_EN_24_TOTAL_BELOW(I) -
MAIN_EN_24_LOW2_BELOW(I);

[EN_24_HIGH_TOT_CALC] MAIN_EN_24_MOD(I) - (1+MAIN_EN_24_HIGH1(I))*MAIN_EN_24_NAT(I) =
MAIN_EN_24_TOTAL_ABOVE(I) - MAIN_EN_24_HIGH1_BELOW(I);
[EN_24_HIGH2_CALC] MAIN_EN_24_MOD(I) - (1+MAIN_EN_24_HIGH2(I))*MAIN_EN_24_NAT(I) =
MAIN_EN_24_HIGH2_ABOVE(I) - MAIN_EN_24_HIGH2_BELOW(I);
[EN_24_HIGH1_CALC] MAIN_EN_24_HIGH1_ABOVE(I) = MAIN_EN_24_TOTAL_ABOVE(I) -
MAIN_EN_24_HIGH2_ABOVE(I);


[EN_26_LOW2_CALC]  MAIN_EN_26_MOD(I) - (1+MAIN_EN_26_LOW2(I))*MAIN_EN_26_NAT(I) =
MAIN_EN_26_LOW2_ABOVE(I) - MAIN_EN_26_LOW2_BELOW(I);
[EN_26_LOW_TOT_CALC]  MAIN_EN_26_MOD(I) - (1+MAIN_EN_26_LOW1(I))*MAIN_EN_26_NAT(I) =
MAIN_EN_26_LOW1_ABOVE(I) - MAIN_EN_26_TOTAL_BELOW(I);
[EN_26_LOW1_CALC]  MAIN_EN_26_LOW1_BELOW(I) = MAIN_EN_26_TOTAL_BELOW(I) -
MAIN_EN_26_LOW2_BELOW(I);

[EN_26_HIGH_TOT_CALC] MAIN_EN_26_MOD(I) - (1+MAIN_EN_26_HIGH1(I))*MAIN_EN_26_NAT(I) =
MAIN_EN_26_TOTAL_ABOVE(I) - MAIN_EN_26_HIGH1_BELOW(I);
[EN_26_HIGH2_CALC] MAIN_EN_26_MOD(I) - (1+MAIN_EN_26_HIGH2(I))*MAIN_EN_26_NAT(I) =
MAIN_EN_26_HIGH2_ABOVE(I) - MAIN_EN_26_HIGH2_BELOW(I);
[EN_26_HIGH1_CALC] MAIN_EN_26_HIGH1_ABOVE(I) = MAIN_EN_26_TOTAL_ABOVE(I) -
MAIN_EN_26_HIGH2_ABOVE(I);


[EN_33_LOW2_CALC]  WST_EN_33_MOD(I) - (1+WST_EN_33_LOW2(I))*WST_EN_33_NAT(I) =
WST_EN_33_LOW2_ABOVE(I) - WST_EN_33_LOW2_BELOW(I);
[EN_33_LOW_TOT_CALC]  WST_EN_33_MOD(I) - (1+WST_EN_33_LOW1(I))*WST_EN_33_NAT(I) =
WST_EN_33_LOW1_ABOVE(I) - WST_EN_33_TOTAL_BELOW(I);
[EN_33_LOW1_CALC]  WST_EN_33_LOW1_BELOW(I) = WST_EN_33_TOTAL_BELOW(I) -
WST_EN_33_LOW2_BELOW(I);

[EN_33_HIGH_TOT_CALC] WST_EN_33_MOD(I) - (1+WST_EN_33_HIGH1(I))*WST_EN_33_NAT(I) =
WST_EN_33_TOTAL_ABOVE(I) - WST_EN_33_HIGH1_BELOW(I);
[EN_33_HIGH2_CALC] WST_EN_33_MOD(I) - (1+WST_EN_33_HIGH2(I))*WST_EN_33_NAT(I) =
WST_EN_33_HIGH2_ABOVE(I) - WST_EN_33_HIGH2_BELOW(I);
```

```
[EN_33_HIGH1_CALC] WST_EN_33_HIGH1_ABOVE(I) = WST_EN_33_TOTAL_ABOVE(I) -
WST_EN_33_HIGH2_ABOVE(I);


[EN_37_LOW2_CALC]  ASH_EN_37_MOD(I) - (1+ASH_EN_37_LOW2(I))*ASH_EN_37_NAT(I) =
ASH_EN_37_LOW2_ABOVE(I) - ASH_EN_37_LOW2_BELOW(I);
[EN_37_LOW_TOT_CALC]  ASH_EN_37_MOD(I) - (1+ASH_EN_37_LOW1(I))*ASH_EN_37_NAT(I) =
ASH_EN_37_LOW1_ABOVE(I) - ASH_EN_37_TOTAL_BELOW(I);
[EN_37_LOW1_CALC]  ASH_EN_37_LOW1_BELOW(I) = ASH_EN_37_TOTAL_BELOW(I) -
ASH_EN_37_LOW2_BELOW(I);

[EN_37_HIGH_TOT_CALC] ASH_EN_37_MOD(I) - (1+ASH_EN_37_HIGH1(I))*ASH_EN_37_NAT(I) =
ASH_EN_37_TOTAL_ABOVE(I) - ASH_EN_37_HIGH1_BELOW(I);
[EN_37_HIGH2_CALC] ASH_EN_37_MOD(I) - (1+ASH_EN_37_HIGH2(I))*ASH_EN_37_NAT(I) =
ASH_EN_37_HIGH2_ABOVE(I) - ASH_EN_37_HIGH2_BELOW(I);
[EN_37_HIGH1_CALC] ASH_EN_37_HIGH1_ABOVE(I) = ASH_EN_37_TOTAL_ABOVE(I) -
ASH_EN_37_HIGH2_ABOVE(I);


[EN_39_LOW2_CALC]  MLR_EN_39_MOD(I) - (1+MLR_EN_39_LOW2(I))*MLR_EN_39_NAT(I) =
MLR_EN_39_LOW2_ABOVE(I) - MLR_EN_39_LOW2_BELOW(I);
[EN_39_LOW_TOT_CALC]  MLR_EN_39_MOD(I) - (1+MLR_EN_39_LOW1(I))*MLR_EN_39_NAT(I) =
MLR_EN_39_LOW1_ABOVE(I) - MLR_EN_39_TOTAL_BELOW(I);
[EN_39_LOW1_CALC]  MLR_EN_39_LOW1_BELOW(I) = MLR_EN_39_TOTAL_BELOW(I) -
MLR_EN_39_LOW2_BELOW(I);

[EN_39_HIGH_TOT_CALC] MLR_EN_39_MOD(I) - (1+MLR_EN_39_HIGH1(I))*MLR_EN_39_NAT(I) =
MLR_EN_39_TOTAL_ABOVE(I) - MLR_EN_39_HIGH1_BELOW(I);
[EN_39_HIGH2_CALC] MLR_EN_39_MOD(I) - (1+MLR_EN_39_HIGH2(I))*MLR_EN_39_NAT(I) =
MLR_EN_39_HIGH2_ABOVE(I) - MLR_EN_39_HIGH2_BELOW(I);
[EN_39_HIGH1_CALC] MLR_EN_39_HIGH1_ABOVE(I) = MLR_EN_39_TOTAL_ABOVE(I) -
MLR_EN_39_HIGH2_ABOVE(I);


[EN_45_LOW2_CALC]  DRF_EN_45_MOD(I) - (1+DRF_EN_45_LOW2(I))*DRF_EN_45_NAT(I) =
DRF_EN_45_LOW2_ABOVE(I) - DRF_EN_45_LOW2_BELOW(I);
[EN_45_LOW_TOT_CALC]  DRF_EN_45_MOD(I) - (1+DRF_EN_45_LOW1(I))*DRF_EN_45_NAT(I) =
DRF_EN_45_LOW1_ABOVE(I) - DRF_EN_45_TOTAL_BELOW(I);
[EN_45_LOW1_CALC]  DRF_EN_45_LOW1_BELOW(I) = DRF_EN_45_TOTAL_BELOW(I) -
DRF_EN_45_LOW2_BELOW(I);

[EN_45_HIGH_TOT_CALC] DRF_EN_45_MOD(I) - (1+DRF_EN_45_HIGH1(I))*DRF_EN_45_NAT(I) =
DRF_EN_45_TOTAL_ABOVE(I) - DRF_EN_45_HIGH1_BELOW(I);
[EN_45_HIGH2_CALC] DRF_EN_45_MOD(I) - (1+DRF_EN_45_HIGH2(I))*DRF_EN_45_NAT(I) =
DRF_EN_45_HIGH2_ABOVE(I) - DRF_EN_45_HIGH2_BELOW(I);
[EN_45_HIGH1_CALC] DRF_EN_45_HIGH1_ABOVE(I) = DRF_EN_45_TOTAL_ABOVE(I) -
DRF_EN_45_HIGH2_ABOVE(I);


[EN_59_LOW2_CALC]  FAR_EN_59_MOD(I) - (1+FAR_EN_59_LOW2(I))*FAR_EN_59_NAT(I) =
FAR_EN_59_LOW2_ABOVE(I) - FAR_EN_59_LOW2_BELOW(I);
[EN_59_LOW_TOT_CALC]  FAR_EN_59_MOD(I) - (1+FAR_EN_59_LOW1(I))*FAR_EN_59_NAT(I) =
FAR_EN_59_LOW1_ABOVE(I) - FAR_EN_59_TOTAL_BELOW(I);
[EN_59_LOW1_CALC]  FAR_EN_59_LOW1_BELOW(I) = FAR_EN_59_TOTAL_BELOW(I) -
FAR_EN_59_LOW2_BELOW(I);

[EN_59_HIGH_TOT_CALC] FAR_EN_59_MOD(I) - (1+FAR_EN_59_HIGH1(I))*FAR_EN_59_NAT(I) =
FAR_EN_59_TOTAL_ABOVE(I) - FAR_EN_59_HIGH1_BELOW(I);
[EN_59_HIGH2_CALC] FAR_EN_59_MOD(I) - (1+FAR_EN_59_HIGH2(I))*FAR_EN_59_NAT(I) =
FAR_EN_59_HIGH2_ABOVE(I) - FAR_EN_59_HIGH2_BELOW(I);
[EN_59_HIGH1_CALC] FAR_EN_59_HIGH1_ABOVE(I) = FAR_EN_59_TOTAL_ABOVE(I) -
FAR_EN_59_HIGH2_ABOVE(I);
```

```
[EN_74_LOW2_CALC]  MAIN_EN_74_MOD(I) - (1+MAIN_EN_74_LOW2(I))*MAIN_EN_74_NAT(I) =
MAIN_EN_74_LOW2_ABOVE(I) - MAIN_EN_74_LOW2_BELOW(I);
[EN_74_LOW_TOT_CALC]  MAIN_EN_74_MOD(I) - (1+MAIN_EN_74_LOW1(I))*MAIN_EN_74_NAT(I) =
MAIN_EN_74_LOW1_ABOVE(I) - MAIN_EN_74_TOTAL_BELOW(I);
[EN_74_LOW1_CALC]  MAIN_EN_74_LOW1_BELOW(I) = MAIN_EN_74_TOTAL_BELOW(I) -
MAIN_EN_74_LOW2_BELOW(I);

[EN_74_HIGH_TOT_CALC] MAIN_EN_74_MOD(I) - (1+MAIN_EN_74_HIGH1(I))*MAIN_EN_74_NAT(I) =
MAIN_EN_74_TOTAL_ABOVE(I) - MAIN_EN_74_HIGH1_BELOW(I);
[EN_74_HIGH2_CALC] MAIN_EN_74_MOD(I) - (1+MAIN_EN_74_HIGH2(I))*MAIN_EN_74_NAT(I) =
MAIN_EN_74_HIGH2_ABOVE(I) - MAIN_EN_74_HIGH2_BELOW(I);
[EN_74_HIGH1_CALC] MAIN_EN_74_HIGH1_ABOVE(I) = MAIN_EN_74_TOTAL_ABOVE(I) -
MAIN_EN_74_HIGH2_ABOVE(I);


[EN_76_LOW2_CALC]  WSF_EN_76_MOD(I) - (1+WSF_EN_76_LOW2(I))*WSF_EN_76_NAT(I) =
WSF_EN_76_LOW2_ABOVE(I) - WSF_EN_76_LOW2_BELOW(I);
[EN_76_LOW_TOT_CALC]  WSF_EN_76_MOD(I) - (1+WSF_EN_76_LOW1(I))*WSF_EN_76_NAT(I) =
WSF_EN_76_LOW1_ABOVE(I) - WSF_EN_76_TOTAL_BELOW(I);
[EN_76_LOW1_CALC]  WSF_EN_76_LOW1_BELOW(I) = WSF_EN_76_TOTAL_BELOW(I) -
WSF_EN_76_LOW2_BELOW(I);

[EN_76_HIGH_TOT_CALC] WSF_EN_76_MOD(I) - (1+WSF_EN_76_HIGH1(I))*WSF_EN_76_NAT(I) =
WSF_EN_76_TOTAL_ABOVE(I) - WSF_EN_76_HIGH1_BELOW(I);
[EN_76_HIGH2_CALC] WSF_EN_76_MOD(I) - (1+WSF_EN_76_HIGH2(I))*WSF_EN_76_NAT(I) =
WSF_EN_76_HIGH2_ABOVE(I) - WSF_EN_76_HIGH2_BELOW(I);
[EN_76_HIGH1_CALC] WSF_EN_76_HIGH1_ABOVE(I) = WSF_EN_76_TOTAL_ABOVE(I) -
WSF_EN_76_HIGH2_ABOVE(I);


[EN_78_LOW2_CALC]  WSF_EN_78_MOD(I) - (1+WSF_EN_78_LOW2(I))*WSF_EN_78_NAT(I) =
WSF_EN_78_LOW2_ABOVE(I) - WSF_EN_78_LOW2_BELOW(I);
[EN_78_LOW_TOT_CALC]  WSF_EN_78_MOD(I) - (1+WSF_EN_78_LOW1(I))*WSF_EN_78_NAT(I) =
WSF_EN_78_LOW1_ABOVE(I) - WSF_EN_78_TOTAL_BELOW(I);
[EN_78_LOW1_CALC]  WSF_EN_78_LOW1_BELOW(I) = WSF_EN_78_TOTAL_BELOW(I) -
WSF_EN_78_LOW2_BELOW(I);

[EN_78_HIGH_TOT_CALC] WSF_EN_78_MOD(I) - (1+WSF_EN_78_HIGH1(I))*WSF_EN_78_NAT(I) =
WSF_EN_78_TOTAL_ABOVE(I) - WSF_EN_78_HIGH1_BELOW(I);
[EN_78_HIGH2_CALC] WSF_EN_78_MOD(I) - (1+WSF_EN_78_HIGH2(I))*WSF_EN_78_NAT(I) =
WSF_EN_78_HIGH2_ABOVE(I) - WSF_EN_78_HIGH2_BELOW(I);
[EN_78_HIGH1_CALC] WSF_EN_78_HIGH1_ABOVE(I) = WSF_EN_78_TOTAL_ABOVE(I) -
WSF_EN_78_HIGH2_ABOVE(I);


[EN_80_LOW2_CALC]  CHP_EN_80_MOD(I) - (1+CHP_EN_80_LOW2(I))*CHP_EN_80_NAT(I) =
CHP_EN_80_LOW2_ABOVE(I) - CHP_EN_80_LOW2_BELOW(I);
[EN_80_LOW_TOT_CALC]  CHP_EN_80_MOD(I) - (1+CHP_EN_80_LOW1(I))*CHP_EN_80_NAT(I) =
CHP_EN_80_LOW1_ABOVE(I) - CHP_EN_80_TOTAL_BELOW(I);
[EN_80_LOW1_CALC]  CHP_EN_80_LOW1_BELOW(I) = CHP_EN_80_TOTAL_BELOW(I) -
CHP_EN_80_LOW2_BELOW(I);

[EN_80_HIGH_TOT_CALC] CHP_EN_80_MOD(I) - (1+CHP_EN_80_HIGH1(I))*CHP_EN_80_NAT(I) =
CHP_EN_80_TOTAL_ABOVE(I) - CHP_EN_80_HIGH1_BELOW(I);
[EN_80_HIGH2_CALC] CHP_EN_80_MOD(I) - (1+CHP_EN_80_HIGH2(I))*CHP_EN_80_NAT(I) =
CHP_EN_80_HIGH2_ABOVE(I) - CHP_EN_80_HIGH2_BELOW(I);
[EN_80_HIGH1_CALC] CHP_EN_80_HIGH1_ABOVE(I) = CHP_EN_80_TOTAL_ABOVE(I) -
CHP_EN_80_HIGH2_ABOVE(I);


[EN_88_LOW2_CALC]  MAIN_EN_88_MOD(I) - (1+MAIN_EN_88_LOW2(I))*MAIN_EN_88_NAT(I) =
MAIN_EN_88_LOW2_ABOVE(I) - MAIN_EN_88_LOW2_BELOW(I);
```

```
[EN_88_LOW_TOT_CALC]  MAIN_EN_88_MOD(I) - (1+MAIN_EN_88_LOW1(I))*MAIN_EN_88_NAT(I) =
MAIN_EN_88_LOW1_ABOVE(I) - MAIN_EN_88_TOTAL_BELOW(I);
[EN_88_LOW1_CALC]  MAIN_EN_88_LOW1_BELOW(I) = MAIN_EN_88_TOTAL_BELOW(I) -
MAIN_EN_88_LOW2_BELOW(I);

[EN_88_HIGH_TOT_CALC] MAIN_EN_88_MOD(I) - (1+MAIN_EN_88_HIGH1(I))*MAIN_EN_88_NAT(I) =
MAIN_EN_88_TOTAL_ABOVE(I) - MAIN_EN_88_HIGH1_BELOW(I);
[EN_88_HIGH2_CALC] MAIN_EN_88_MOD(I) - (1+MAIN_EN_88_HIGH2(I))*MAIN_EN_88_NAT(I) =
MAIN_EN_88_HIGH2_ABOVE(I) - MAIN_EN_88_HIGH2_BELOW(I);
[EN_88_HIGH1_CALC] MAIN_EN_88_HIGH1_ABOVE(I) = MAIN_EN_88_TOTAL_ABOVE(I) -
MAIN_EN_88_HIGH2_ABOVE(I);


[EN_91_LOW2_CALC]  ASH_EN_91_MOD(I) - (1+ASH_EN_91_LOW2(I))*ASH_EN_91_NAT(I) =
ASH_EN_91_LOW2_ABOVE(I) - ASH_EN_91_LOW2_BELOW(I);
[EN_91_LOW_TOT_CALC]  ASH_EN_91_MOD(I) - (1+ASH_EN_91_LOW1(I))*ASH_EN_91_NAT(I) =
ASH_EN_91_LOW1_ABOVE(I) - ASH_EN_91_TOTAL_BELOW(I);
[EN_91_LOW1_CALC]  ASH_EN_91_LOW1_BELOW(I) = ASH_EN_91_TOTAL_BELOW(I) -
ASH_EN_91_LOW2_BELOW(I);

[EN_91_HIGH_TOT_CALC] ASH_EN_91_MOD(I) - (1+ASH_EN_91_HIGH1(I))*ASH_EN_91_NAT(I) =
ASH_EN_91_TOTAL_ABOVE(I) - ASH_EN_91_HIGH1_BELOW(I);
[EN_91_HIGH2_CALC] ASH_EN_91_MOD(I) - (1+ASH_EN_91_HIGH2(I))*ASH_EN_91_NAT(I) =
ASH_EN_91_HIGH2_ABOVE(I) - ASH_EN_91_HIGH2_BELOW(I);
[EN_91_HIGH1_CALC] ASH_EN_91_HIGH1_ABOVE(I) = ASH_EN_91_TOTAL_ABOVE(I) -
ASH_EN_91_HIGH2_ABOVE(I);


[EN_95_LOW2_CALC]  WST_EN_95_MOD(I) - (1+WST_EN_95_LOW2(I))*WST_EN_95_NAT(I) =
WST_EN_95_LOW2_ABOVE(I) - WST_EN_95_LOW2_BELOW(I);
[EN_95_LOW_TOT_CALC]  WST_EN_95_MOD(I) - (1+WST_EN_95_LOW1(I))*WST_EN_95_NAT(I) =
WST_EN_95_LOW1_ABOVE(I) - WST_EN_95_TOTAL_BELOW(I);
[EN_95_LOW1_CALC]  WST_EN_95_LOW1_BELOW(I) = WST_EN_95_TOTAL_BELOW(I) -
WST_EN_95_LOW2_BELOW(I);

[EN_95_HIGH_TOT_CALC] WST_EN_95_MOD(I) - (1+WST_EN_95_HIGH1(I))*WST_EN_95_NAT(I) =
WST_EN_95_TOTAL_ABOVE(I) - WST_EN_95_HIGH1_BELOW(I);
[EN_95_HIGH2_CALC] WST_EN_95_MOD(I) - (1+WST_EN_95_HIGH2(I))*WST_EN_95_NAT(I) =
WST_EN_95_HIGH2_ABOVE(I) - WST_EN_95_HIGH2_BELOW(I);
[EN_95_HIGH1_CALC] WST_EN_95_HIGH1_ABOVE(I) = WST_EN_95_TOTAL_ABOVE(I) -
WST_EN_95_HIGH2_ABOVE(I);


[EN_98_LOW2_CALC]  MAIN_EN_98_MOD(I) - (1+MAIN_EN_98_LOW2(I))*MAIN_EN_98_NAT(I) =
MAIN_EN_98_LOW2_ABOVE(I) - MAIN_EN_98_LOW2_BELOW(I);
[EN_98_LOW_TOT_CALC]  MAIN_EN_98_MOD(I) - (1+MAIN_EN_98_LOW1(I))*MAIN_EN_98_NAT(I) =
MAIN_EN_98_LOW1_ABOVE(I) - MAIN_EN_98_TOTAL_BELOW(I);
[EN_98_LOW1_CALC]  MAIN_EN_98_LOW1_BELOW(I) = MAIN_EN_98_TOTAL_BELOW(I) -
MAIN_EN_98_LOW2_BELOW(I);

[EN_98_HIGH_TOT_CALC] MAIN_EN_98_MOD(I) - (1+MAIN_EN_98_HIGH1(I))*MAIN_EN_98_NAT(I) =
MAIN_EN_98_TOTAL_ABOVE(I) - MAIN_EN_98_HIGH1_BELOW(I);
[EN_98_HIGH2_CALC] MAIN_EN_98_MOD(I) - (1+MAIN_EN_98_HIGH2(I))*MAIN_EN_98_NAT(I) =
MAIN_EN_98_HIGH2_ABOVE(I) - MAIN_EN_98_HIGH2_BELOW(I);
[EN_98_HIGH1_CALC] MAIN_EN_98_HIGH1_ABOVE(I) = MAIN_EN_98_TOTAL_ABOVE(I) -
MAIN_EN_98_HIGH2_ABOVE(I);


[EN_100_LOW2_CALC]  MAIN_EN_100_MOD(I) - (1+MAIN_EN_100_LOW2(I))*MAIN_EN_100_NAT(I) =
MAIN_EN_100_LOW2_ABOVE(I) - MAIN_EN_100_LOW2_BELOW(I);
[EN_100_LOW_TOT_CALC]  MAIN_EN_100_MOD(I) - (1+MAIN_EN_100_LOW1(I))*MAIN_EN_100_NAT(I)
= MAIN_EN_100_LOW1_ABOVE(I) - MAIN_EN_100_TOTAL_BELOW(I);
```

```
[EN_100_LOW1_CALC]  MAIN_EN_100_LOW1_BELOW(I) = MAIN_EN_100_TOTAL_BELOW(I) -
MAIN_EN_100_LOW2_BELOW(I);

[EN_100_HIGH_TOT_CALC] MAIN_EN_100_MOD(I) -
(1+MAIN_EN_100_HIGH1(I))*MAIN_EN_100_NAT(I) = MAIN_EN_100_TOTAL_ABOVE(I) -
MAIN_EN_100_HIGH1_BELOW(I);
[EN_100_HIGH2_CALC] MAIN_EN_100_MOD(I) - (1+MAIN_EN_100_HIGH2(I))*MAIN_EN_100_NAT(I) =
MAIN_EN_100_HIGH2_ABOVE(I) - MAIN_EN_100_HIGH2_BELOW(I);
[EN_100_HIGH1_CALC] MAIN_EN_100_HIGH1_ABOVE(I) = MAIN_EN_100_TOTAL_ABOVE(I) -
MAIN_EN_100_HIGH2_ABOVE(I);


[EN_136_LOW2_CALC]  DRF_EN_136_MOD(I) - (1+DRF_EN_136_LOW2(I))*DRF_EN_136_NAT(I) =
DRF_EN_136_LOW2_ABOVE(I) - DRF_EN_136_LOW2_BELOW(I);
[EN_136_LOW_TOT_CALC]  DRF_EN_136_MOD(I) - (1+DRF_EN_136_LOW1(I))*DRF_EN_136_NAT(I) =
DRF_EN_136_LOW1_ABOVE(I) - DRF_EN_136_TOTAL_BELOW(I);
[EN_136_LOW1_CALC]  DRF_EN_136_LOW1_BELOW(I) = DRF_EN_136_TOTAL_BELOW(I) -
DRF_EN_136_LOW2_BELOW(I);

[EN_136_HIGH_TOT_CALC] DRF_EN_136_MOD(I) - (1+DRF_EN_136_HIGH1(I))*DRF_EN_136_NAT(I) =
DRF_EN_136_TOTAL_ABOVE(I) - DRF_EN_136_HIGH1_BELOW(I);
[EN_136_HIGH2_CALC] DRF_EN_136_MOD(I) - (1+DRF_EN_136_HIGH2(I))*DRF_EN_136_NAT(I) =
DRF_EN_136_HIGH2_ABOVE(I) - DRF_EN_136_HIGH2_BELOW(I);
[EN_136_HIGH1_CALC] DRF_EN_136_HIGH1_ABOVE(I) = DRF_EN_136_TOTAL_ABOVE(I) -
DRF_EN_136_HIGH2_ABOVE(I);


[EN_137_LOW2_CALC]  DRF_EN_137_MOD(I) - (1+DRF_EN_137_LOW2(I))*DRF_EN_137_NAT(I) =
DRF_EN_137_LOW2_ABOVE(I) - DRF_EN_137_LOW2_BELOW(I);
[EN_137_LOW_TOT_CALC]  DRF_EN_137_MOD(I) - (1+DRF_EN_137_LOW1(I))*DRF_EN_137_NAT(I) =
DRF_EN_137_LOW1_ABOVE(I) - DRF_EN_137_TOTAL_BELOW(I);
[EN_137_LOW1_CALC]  DRF_EN_137_LOW1_BELOW(I) = DRF_EN_137_TOTAL_BELOW(I) -
DRF_EN_137_LOW2_BELOW(I);

[EN_137_HIGH_TOT_CALC] DRF_EN_137_MOD(I) - (1+DRF_EN_137_HIGH1(I))*DRF_EN_137_NAT(I) =
DRF_EN_137_TOTAL_ABOVE(I) - DRF_EN_137_HIGH1_BELOW(I);
[EN_137_HIGH2_CALC] DRF_EN_137_MOD(I) - (1+DRF_EN_137_HIGH2(I))*DRF_EN_137_NAT(I) =
DRF_EN_137_HIGH2_ABOVE(I) - DRF_EN_137_HIGH2_BELOW(I);
[EN_137_HIGH1_CALC] DRF_EN_137_HIGH1_ABOVE(I) = DRF_EN_137_TOTAL_ABOVE(I) -
DRF_EN_137_HIGH2_ABOVE(I);


[EN_139_LOW2_CALC]  FAR_EN_139_MOD(I) - (1+FAR_EN_139_LOW2(I))*FAR_EN_139_NAT(I) =
FAR_EN_139_LOW2_ABOVE(I) - FAR_EN_139_LOW2_BELOW(I);
[EN_139_LOW_TOT_CALC]  FAR_EN_139_MOD(I) - (1+FAR_EN_139_LOW1(I))*FAR_EN_139_NAT(I) =
FAR_EN_139_LOW1_ABOVE(I) - FAR_EN_139_TOTAL_BELOW(I);
[EN_139_LOW1_CALC]  FAR_EN_139_LOW1_BELOW(I) = FAR_EN_139_TOTAL_BELOW(I) -
FAR_EN_139_LOW2_BELOW(I);

[EN_139_HIGH_TOT_CALC] FAR_EN_139_MOD(I) - (1+FAR_EN_139_HIGH1(I))*FAR_EN_139_NAT(I) =
FAR_EN_139_TOTAL_ABOVE(I) - FAR_EN_139_HIGH1_BELOW(I);
[EN_139_HIGH2_CALC] FAR_EN_139_MOD(I) - (1+FAR_EN_139_HIGH2(I))*FAR_EN_139_NAT(I) =
FAR_EN_139_HIGH2_ABOVE(I) - FAR_EN_139_HIGH2_BELOW(I);
[EN_139_HIGH1_CALC] FAR_EN_139_HIGH1_ABOVE(I) = FAR_EN_139_TOTAL_ABOVE(I) -
FAR_EN_139_HIGH2_ABOVE(I);


[EN_149_LOW2_CALC]  FAR_EN_149_MOD(I) - (1+FAR_EN_149_LOW2(I))*FAR_EN_149_NAT(I) =
FAR_EN_149_LOW2_ABOVE(I) - FAR_EN_149_LOW2_BELOW(I);
[EN_149_LOW_TOT_CALC]  FAR_EN_149_MOD(I) - (1+FAR_EN_149_LOW1(I))*FAR_EN_149_NAT(I) =
FAR_EN_149_LOW1_ABOVE(I) - FAR_EN_149_TOTAL_BELOW(I);
[EN_149_LOW1_CALC]  FAR_EN_149_LOW1_BELOW(I) = FAR_EN_149_TOTAL_BELOW(I) -
FAR_EN_149_LOW2_BELOW(I);
```

```
[EN_149_HIGH_TOT_CALC] FAR_EN_149_MOD(I) - (1+FAR_EN_149_HIGH1(I))*FAR_EN_149_NAT(I) =
FAR_EN_149_TOTAL_ABOVE(I) - FAR_EN_149_HIGH1_BELOW(I);
[EN_149_HIGH2_CALC] FAR_EN_149_MOD(I) - (1+FAR_EN_149_HIGH2(I))*FAR_EN_149_NAT(I) =
FAR_EN_149_HIGH2_ABOVE(I) - FAR_EN_149_HIGH2_BELOW(I);
[EN_149_HIGH1_CALC] FAR_EN_149_HIGH1_ABOVE(I) = FAR_EN_149_TOTAL_ABOVE(I) -
FAR_EN_149_HIGH2_ABOVE(I);


[EN_152_LOW2_CALC]  MAIN_EN_152_MOD(I) - (1+MAIN_EN_152_LOW2(I))*MAIN_EN_152_NAT(I) =
MAIN_EN_152_LOW2_ABOVE(I) - MAIN_EN_152_LOW2_BELOW(I);
[EN_152_LOW_TOT_CALC]  MAIN_EN_152_MOD(I) - (1+MAIN_EN_152_LOW1(I))*MAIN_EN_152_NAT(I)
= MAIN_EN_152_LOW1_ABOVE(I) - MAIN_EN_152_TOTAL_BELOW(I);
[EN_152_LOW1_CALC]  MAIN_EN_152_LOW1_BELOW(I) = MAIN_EN_152_TOTAL_BELOW(I) -
MAIN_EN_152_LOW2_BELOW(I);

[EN_152_HIGH_TOT_CALC] MAIN_EN_152_MOD(I) -
(1+MAIN_EN_152_HIGH1(I))*MAIN_EN_152_NAT(I) = MAIN_EN_152_TOTAL_ABOVE(I) -
MAIN_EN_152_HIGH1_BELOW(I);
[EN_152_HIGH2_CALC] MAIN_EN_152_MOD(I) - (1+MAIN_EN_152_HIGH2(I))*MAIN_EN_152_NAT(I) =
MAIN_EN_152_HIGH2_ABOVE(I) - MAIN_EN_152_HIGH2_BELOW(I);
[EN_152_HIGH1_CALC] MAIN_EN_152_HIGH1_ABOVE(I) = MAIN_EN_152_TOTAL_ABOVE(I) -
MAIN_EN_152_HIGH2_ABOVE(I);

);


!TURN ECO TARGETS INTO CONSTRAINTS;
!@SUM (day(I):
(MAIN_EN_1_LOW1_BELOW(I) + MAIN_EN_1_HIGH1_ABOVE(I) +
MAIN_EN_6_LOW1_BELOW(I) + MAIN_EN_6_HIGH1_ABOVE(I) +
MAIN_EN_9_LOW1_BELOW(I) + MAIN_EN_9_HIGH1_ABOVE(I) +
MAIN_EN_11_LOW1_BELOW(I) + MAIN_EN_11_HIGH1_ABOVE(I) +
MAS_EN_19_LOW1_BELOW(I) + MAS_EN_19_HIGH1_ABOVE(I) +
BLK_EN_21_LOW1_BELOW(I) + BLK_EN_21_HIGH1_ABOVE(I) +
SGR_EN_22_LOW1_BELOW(I) + SGR_EN_22_HIGH1_ABOVE(I) +
MAIN_EN_24_LOW1_BELOW(I) + MAIN_EN_24_HIGH1_ABOVE(I) +
MAIN_EN_26_LOW1_BELOW(I) + MAIN_EN_26_HIGH1_ABOVE(I) +
WST_EN_33_LOW1_BELOW(I) + WST_EN_33_HIGH1_ABOVE(I) +
ASH_EN_37_LOW1_BELOW(I) + ASH_EN_37_HIGH1_ABOVE(I) +
MLR_EN_39_LOW1_BELOW(I) + MLR_EN_39_HIGH1_ABOVE(I) +
DRF_EN_45_LOW1_BELOW(I) + DRF_EN_45_HIGH1_ABOVE(I) +
FAR_EN_59_LOW1_BELOW(I) + FAR_EN_59_HIGH1_ABOVE(I) +
MAIN_EN_74_LOW1_BELOW(I) + MAIN_EN_74_HIGH1_ABOVE(I) +
WSF_EN_76_LOW1_BELOW(I) + WSF_EN_76_HIGH1_ABOVE(I) +
WSF_EN_78_LOW1_BELOW(I) + WSF_EN_78_HIGH1_ABOVE(I) +
CHP_EN_80_LOW1_BELOW(I) + CHP_EN_80_HIGH1_ABOVE(I) +
MAIN_EN_88_LOW1_BELOW(I) + MAIN_EN_88_HIGH1_ABOVE(I) +
ASH_EN_91_LOW1_BELOW(I) + ASH_EN_91_HIGH1_ABOVE(I) +
WST_EN_95_LOW1_BELOW(I) + WST_EN_95_HIGH1_ABOVE(I) +
MAIN_EN_98_LOW1_BELOW(I) + MAIN_EN_98_HIGH1_ABOVE(I) +
MAIN_EN_100_LOW1_BELOW(I) + MAIN_EN_100_HIGH1_ABOVE(I) +
DRF_EN_136_LOW1_BELOW(I) + DRF_EN_136_HIGH1_ABOVE(I) +
DRF_EN_137_LOW1_BELOW(I) + DRF_EN_137_HIGH1_ABOVE(I) +
FAR_EN_139_LOW1_BELOW(I) + FAR_EN_139_HIGH1_ABOVE(I) +
FAR_EN_149_LOW1_BELOW(I) + FAR_EN_149_HIGH1_ABOVE(I) +
MAIN_EN_152_LOW1_BELOW(I) + MAIN_EN_152_HIGH1_ABOVE(I))

+2*(
MAIN_EN_1_LOW2_BELOW(I) + MAIN_EN_1_HIGH2_ABOVE(I) +
MAIN_EN_6_LOW2_BELOW(I) + MAIN_EN_6_HIGH2_ABOVE(I) +
MAIN_EN_9_LOW2_BELOW(I) + MAIN_EN_9_HIGH2_ABOVE(I) +
MAIN_EN_11_LOW2_BELOW(I) + MAIN_EN_11_HIGH2_ABOVE(I) +
```

```
MAS_EN_19_LOW2_BELOW(I) + MAS_EN_19_HIGH2_ABOVE(I) +
BLK_EN_21_LOW2_BELOW(I) + BLK_EN_21_HIGH2_ABOVE(I) +
SGR_EN_22_LOW2_BELOW(I) + SGR_EN_22_HIGH2_ABOVE(I) +
MAIN_EN_24_LOW2_BELOW(I) + MAIN_EN_24_HIGH2_ABOVE(I) +
MAIN_EN_26_LOW2_BELOW(I) + MAIN_EN_26_HIGH2_ABOVE(I) +
WST_EN_33_LOW2_BELOW(I) + WST_EN_33_HIGH2_ABOVE(I) +
ASH_EN_37_LOW2_BELOW(I) + ASH_EN_37_HIGH2_ABOVE(I) +
MLR_EN_39_LOW2_BELOW(I) + MLR_EN_39_HIGH2_ABOVE(I) +
DRF_EN_45_LOW2_BELOW(I) + DRF_EN_45_HIGH2_ABOVE(I) +
FAR_EN_59_LOW2_BELOW(I) + FAR_EN_59_HIGH2_ABOVE(I) +
MAIN_EN_74_LOW2_BELOW(I) + MAIN_EN_74_HIGH2_ABOVE(I) +
WSF_EN_76_LOW2_BELOW(I) + WSF_EN_76_HIGH2_ABOVE(I) +
WSF_EN_78_LOW2_BELOW(I) + WSF_EN_78_HIGH2_ABOVE(I) +
CHP_EN_80_LOW2_BELOW(I) + CHP_EN_80_HIGH2_ABOVE(I) +
MAIN_EN_88_LOW2_BELOW(I) + MAIN_EN_88_HIGH2_ABOVE(I) +
ASH_EN_91_LOW2_BELOW(I) + ASH_EN_91_HIGH2_ABOVE(I) +
WST_EN_95_LOW2_BELOW(I) + WST_EN_95_HIGH2_ABOVE(I) +
MAIN_EN_98_LOW2_BELOW(I) + MAIN_EN_98_HIGH2_ABOVE(I) +
MAIN_EN_100_LOW2_BELOW(I) + MAIN_EN_100_HIGH2_ABOVE(I) +
DRF_EN_136_LOW2_BELOW(I) + DRF_EN_136_HIGH2_ABOVE(I) +
DRF_EN_137_LOW2_BELOW(I) + DRF_EN_137_HIGH2_ABOVE(I) +
FAR_EN_139_LOW2_BELOW(I) + FAR_EN_139_HIGH2_ABOVE(I) +
FAR_EN_149_LOW2_BELOW(I) + FAR_EN_149_HIGH2_ABOVE(I) +
MAIN_EN_152_LOW2_BELOW(I) + MAIN_EN_152_HIGH2_ABOVE(I)
))<=ECO_LIMIT;
```

Optional Hydropower Constraint

This constraint is optionally included to limit hydropower production to a calculated level. This constraint is generally turned off, but can be turned on to limit hydropower production to a level below some defined quantity. This quantity is generally from a specific model run such as a hydropower maximization scenario where hydropower production is weighted to take precedent over other objectives like ecological flows or flood control. Limiting hydropower income through this constraint can be useful in achieving scenarios which demonstrate trade-offs between hydropower and alternative objectives like ecological flows.

```
!HYDROPOWER INCOME LIMIT;
!@SUM(day(I): DRF_TOTAL_INCOME(I) + UP3_TOTAL_INCOME(I) + MID3_TOTAL_INCOME(I) +
CHP_TOTAL_INCOME(I) +
WSF_TOTAL_INCOME(I)+FAR_TOTAL_INCOME(I) + LOW3_TOTAL_INCOME(I)) >
INCOME_LIMIT*Hydropower_Constraint;
```