



## 8.0 Supporting Utilities

This section describes supporting utilities which operate outside the State Model to provide additional plotting and linking capabilities. The following sections are available within this chapter:

- 8.1 [Big Picture Plot](#)
- 8.2 [Basin Linkage](#)
- 8.3 [StateMod File Comparison](#)
- 8.4 [StateDMI](#)
- 8.5 [Tstool DMI](#)
- 8.6 [StateCU](#)
- 8.7 [SmNewRsp](#) (StateMod Response File Program)
- 8.8 [SmDelay](#) (StateMod Delay File Program)

### 8.1 Big Picture Plot

#### Description

The Big Picture Plot is generated by a FORTRAN program named **delplt.f**. **Delplt** post processes one or more output files from StateMod to generate a file which may be viewed as a table or provided to a plotting program to generate a 'Big Picture Plot'. Output from **Delplt** is always directed to the directory where the response file is located. It has the following capabilities:

- Single, Multiple, Difference, Diffx or Merge file results.

The Single option will process the first file only.

The Multiple option will generate a matrix by ID for up to 5 files.

The Difference option will subtract data from two files (ID's in one file but not another will be treated as zeros).

The Diffx option will subtract data from two files (ID's in one file but not another will be ignored).

The Merge option will concatenate two or more files together.

- Operates on both StateMod ASCII and Binary output files.

For ASCII diversion = \*.xdd, reservoir = \*.xre

For Binary diversion - \*.b43, reservoir = \*.b44

- Provides data for one of 20+/- parameters.
- Prints 1, n, or all ID's.
- Prints a specific year, year and month, or average.
- For the Difference option only allows ID's found in one file not in another.

### Constraint:

For the ID Option, the code checks for a -999 as an indicator that no more ID's will be provided.

### Options

The program is written in FORTRAN. It expects a command file which, if not provided, defaults to 'delplt.in'. Following is the format of a command files:

```
Line 1: Run type (Single, Multiple, Difference,  
Merge or Help)  
Line 2: File Name (can be ASCII (e.g. *.xdd) or Binary (e.g. *.b43))  
Line 3: Data Type  
Available data types are  
Diversion  
StreamGage (baseflows)  
Stream (same as StreamGage)  
Reservoir  
Instream  
StreamID (baseflows that begin with a USGS Identifier (e.g. 09... or 08...))  
Line 3: Parameter  
Available Diversion or streamGage or streamID parameters:  
Total_Demand  
CU_Demand  
From_River_by_Priority  
From_River_by_Storage  
From_River_by_Exchange  
From_Well  
From_Carrier_by_Priority  
From_Carrier_by_Storage  
Carried_Water  
From_Soil  
Total_Supply  
Total_Short  
CU_Short  
Consumptive_Use  
To_Soil  
Total_Return  
Loss  
Upstream_Inflow  
Reach_Gain  
Return_Flow  
Well_Depletion  
To/From_GW_Storage  
River_Inflow  
River_Divert
```

River\_by\_Well  
River\_Outflow  
Available\_Flow

Available reservoir parameters:

Initial\_Storage  
River\_Priority  
River\_Storage  
River\_Exchange  
Carrier\_Priority  
Carrier\_Storage  
Total\_Supply  
Storage\_Use  
Storage\_Exchange  
Carrier\_Use  
Total\_Release  
Evap  
Seep\_Spill  
SimEOM  
Target\_Limit  
Fill\_Limit  
River\_Inflow  
Total\_Release  
Total\_Supply  
River\_By\_Well  
River\_Outflow

Line 4: Station ID (0=all, end with a -999)

Line 5: Time (year, year and month, Ave)

### Example of a Difference Application

```
#
# Multiple Files, same data type, same parameter,
#   three years (1975, 1976 and average)
#
#       Run Type: (Single, Multiple, Difference, Merge or Help):
#
Difference
#####
#
#       File:
gunnH.xdd
#
#       Data Type (Diversion, StreamGage, Reservoir, Instream, or
StreamID)
Diversion
#
#       Parameter (same as SMGUI) or type -help
Total_Supply
#
#       ID (0=all, n=ID, end with a -999)
0
-999
#
#       Year or Ave (e.g. Ave or 1989 NOV)
Ave
#
#####
```

```

#
#
#       File:
gunnC.b43
#
#       Data Type (Diversion, StreamGage, Reservoir, Instream, or
StreamID)
Diversion
#
#       Parameter (same as SMGUI) or type -help
Total_Supply
#
#       ID (0=all, n=ID, end with a -999)
0
-999
#
#       Year or Ave (e.g. Ave or 1989 NOV)
Ave
#
-999

```

## 8.2 Basin Linkage

### Description

The Basin Linkage utility, SmLink, allows the input from one or more StateMod input files to be combined in order to operate as a single model. Smlink does the following:

- Reads 2 to 5 StateMod response (\*.rsp) files to generate all the input files required to operate StateMod as a linked basin.
- Allows the user to input replacement commands required to delete nodes where the models overlap or add nodes required to facilitate linkage.
- Generates a log file that records the required dimensions for the StateMod Model and any duplicate ID's that need to be revised before a successful execution of StateMod can be performed.

### Constraints used by the model include:

To link the \*.rin file the code searches for a river node named 'End' or 'END'.

The code warns the user if duplicate ID's are provided in the log file.

The path of each input file is taken from the path specified in the command files unless a path is provided in the response (\*.rsp) file.

The information in the control (\*.ctl) files must be exactly the same (unit conversions, beginning year, etc.) to avoid any warnings. If inconsistent data is provided the information in the first file read is used for the linked control (\*.ctl) file.

Because it is common for several StateMod input files to use the same evaporation, precipitation and delay files the user can specify input data that controls whether or not these files should be linked.

Because it is common for several StateMod input files to use the StateCU input and output files that span an entire watershed the user can specify input data that controls whether or not these files should be linked.

StateMod allows a unit response (\*.urm) or delay \*.dly) file data to be provided in a free format. In order for SmLink to differentiate between an ID and data, columns 1-12 should be reserved for an ID. See below for an example free format unit response file where the ID is URM\_1, 12 unit response values of 10 are provided as input, and columns 1-12 are reserved for an ID designation.

```
URM_1      12 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10.
          10. 10 (data begins in column 13)
```

StateMod allows an operating rule (\*.opr) file to be provided in free format. In order for SmLink to differentiate between an ID and data, columns 1-12 should be reserved for an ID. See below for an example free format operating rule file with monthly on off switch.

```
Opr_Mead.01 Opr_Meadow_D&S_01      100.00000   12.    1
          1 1 1 1 -15 0 0 0 0 0 0 0 (on/off data begins in column 13)
```

## Options

SmLink expects a command file which, if not provided, defaults to 'smlink.rsp'.

SmLink has the capability to perform the following types of edits to an input file

- delsta(fn,id) Delete station ID from file fn
- addrec(fn,rec) Adds a record (rec) to file (fn).
- repzero(fn,id,zero) Set data for station (id) in file (fn) to zero.

## Example

```
#      SmLink.rsp
#
#      Output File name (e.g. wslope)
wslope
#
#
# -----
#      Compare and link selected files
#      (0=no compare and use first file read, 1 yes compare and link files)
#      nEva = evap; nPre = precip; nStr = structure, nIpy = Irrigation Practice,
#      nDly = delay file; nUnit output units = 1 cfs, 2=af, 3=kaf
# nEva nPre nStr nIpy nDdc nDly nUnit
#      0      0      0      0      0      0      3
#
# -----
#      Input File names (e.g. cm2009H.rsp)
\w\statem\Verification\Base\YM2009\ym2009H.rsp
\w\statem\Verification\Base\WM2009\wm2009H.rsp
\w\statem\Verification\Base\SJ2009\SJ2009H2.rsp
```

```

\w\statem\Verification\Base\CM2009\cm2009H2.rsp
\w\statem\Verification\Base\GM2009\gm2009H2.rsp
#
#       River Network
#       09152500 is Gunison R nr Grand Junction
#       420541 is Redlands Power
#       950050 is Redlands Irrig
-delsta(cm2009.rin,09152500)
-delsta(cm2009.rin,420541)
-delsta(gm2009.rin,680636)
#
#       Diversion Stations
-delsta(gm2009.dds,420541)
-delsta(gm2009.dds,680636)
#
#       Add compact node Station, water right and demand
-addrec(gm2009.ifs,Compact_Dem Compact_Dem          Compact_Dem          1
COMPACT_DEM    0          2)
-addrec(gm2009.ifr,Compact_Dem Compact_Dem          Compact_Dem          1.00000
99999.99          1)
-addrec(gm2009.ifa,          Compact_Dem 9999999 9999999 9999999 9999999 9999999
9999999 9999999 9999999 9999999 9999999 9999999 9999999)
#

-repzero(cm2009.rih,950040,zero)
-repzero(cm2009.rih,504600,zero)
-repzero(ym2009.ddh,584686,zero)
-repzero(ym2009H.ddm,584686,zero)
#
-addrec(gm2009.rin,End_File_1  File 1-Compact_Dem  _OTHCompact_Dem  End_File_1
-999)
-addrec(gm2009.rin,End_File_2  File 2-420541      _OTH420541      End_File_2
-999)
-addrec(gm2009.rin,End_File_3  File 3-Compact_Dem  _OTHCompact_Dem  End_File_3
-999)
-addrec(gm2009.rin,End_File_4  File 4-Compact_Dem  _OTHCompact_Dem  End_File_4
-999)
-addrec(gm2009.rin,End_File_5  File 5-Compact_Dem  _OTHCompact_Dem  End_File_5
-999)
-addrec(gm2009.rin,Compact_Dem Compact_Dem          _IFSEnd_All      Compact_Dem
-999)
-addrec(gm2009.rin,End_All      End)

```

## 8.3 StateMod File Comparison

The smfc program is written in FORTRAN and expects a command file with data. Smfc does the following for StateMod applications:

- Reads and compares 2 or more StateMod input or output files. The user has the ability to:

Compare all StateMod input files associated with a run if the file to be compared is a response (\*.rsp) file.

Compare just one file if the file to be compares is anything except a response file (e.g. \*.dds, \*.res, \*.xpl, etc.).

## Constraints used by the model include:

StateMod allows a unit response (\*.urm \*.urd or \*.dly) file data to be provided in a free format. In order for SmFc to differentiate between an ID and data, columns 1-12 of a unit response file should be reserved for an ID. See below for an example free format \*.urm file where the ID is URM\_1, 12 unit response values of 10 are provided as input, and columns 1-12 are reserved for the ID designation.

```
URM_1      12 10. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10.
           10. 10
```

StateMod allows an operating rule (\*.opr) file to be provided in free format. In order for SmLink to differentiate between an ID and data, columns 1-12 should be reserved for an ID. See below for an example free format operating rule file with monthly on off switch.

```
Opr_Mead.01 Opr_Meadow_D&S_01      100.00000   12.    1
          1 1 1 1 -15 0 0 0 0 0 0 0 (on/off data begins in column 13)
```

It is common for a well water right to be assigned more than one structure and a well plan to include many wells. Therefore when comparing a well right file or a well plan file, differences may be expected.

## Options

If a command file is not provided, the program defaults to smfc.rsp.

If a response (\*.rsp) file is provided on the file to compare (line 2) then the code will compare all files contained in the response files.

If any file other than a response if provided then it only compares those two files.

Has the option to print all lines in a comparison (iprint=0) or only lines where differences occur (iprint=1).

Following is the information in a typical command file:

```
Line 1: Print control (iprint) 0=print all; 1=print only delta
Line 2: Comparison control (e.g. .rsp, .dds, etc)
Line 3: Output file name
Line 4: File 1 to compare
Line 5: File 2 to compare
```

### Example

```
#
# Smfc.rsp; response file to smfc.for; StateMod file compare
#
# 1. iprint 0=print all; 1=print only delta
1
#
# 2. Files to compare (.rsp = all)
.rsp
#
```

```
# 3. Output File name
SmFc.out
#
# 4. File 1 to compare
/usr2/crdsswork/statemod/white/whiteH.rsp
#
# 5. File 2 to compare
/usr2/crdsswork/statemod/whiteT/whiteTH.rsp
```

## 8.4 StateDMI

The StateDMI provides the following assistance to the StateMod Model:

- Creates a river network (\*.rin) file that identifies relative location (upstream or downstream) of nodes in a model network.
- Creates a river station (\*.ris) file that describes the names and locations of nodes where baseflows are known.
- Creates a stream estimate station coefficient data (\*.rib) file with proration coefficients to calculate baseflows for nodes where baseflows are not known.
- Creates a reservoir structure (\*.res) file that describes the physical properties of each reservoir in the system.
- Creates a reservoir rights (\*.rer) file that contains data associated with reservoir storage rights.
- Creates an instream flow structure (\*.ifs) file that describes the physical properties of each instream flow in the system.
- Creates an instream flow rights (\*.ifr) file that contains data associated with instream flow water rights.
- Create an instream flow demand (\*.ifa) file that contains annual instream flow demands (12 monthly values) for each instream flow.
- Creates a well structure (\*.wes) file that describes the physical properties of each well in the system.
- Creates a well rights (\*.wer) file that contains data associated with ground water rights.
- Create a well demand (\*.wem) file that contains demands for well structures.
- Creates a direct diversion structure (\*.dds) file that describes the physical properties of each direct diversion in the system.
- Creates a direct diversion rights (\*.ddr) file that contains data associated with diversion water rights.
- Creates an historic diversion (\*.ddh) file by extracting diversion data from the CDSS database.
- Fills missing historic diversion data from user supplied information.



- Calculates the average system efficiency for irrigation structures based on historic diversion data and irrigation water requirement data provided by the CU model, StateCU (see below).
- Creates a demand (\*.ddm) file based on calculated or user supplied efficiency data and farm irrigation water requirement data provided by the CU model, StateCU (see below).
- Extracts irrigated acreage and crop mix (\*.cds) data from the CDSS database.

For a complete description of the StateDMI see CDSS web site (<http://cdss.state.co.us>).

## 8.5 TSTool DMI

The TSTool DMI provides the following assistance to the StateMod Model:

- Extracts historic streamflow data from CDSS database.
- Fills missing streamflow data from user supplied parameters.
- Extracts historic reservoir End-of-Month data from CDSS database.
- Fills missing EOM data from user supplied parameters.
- Extracts precipitation and evaporation data to build the net evaporation file (\*.eva) for StateMod.

For a complete description of the TSTool DMI see CDSS web site (<http://cdss.state.co.us>).

## 8.6 StateCU Model

The StateCU Model provides the following assistance to the StateMod Model:

Provides irrigation water requirement data for estimating irrigation structure efficiencies and calculated demands (as opposed to historic diversions).

For a complete description of the StateCU Model see CDSS web site (<http://cdss.state.co.us>).

## 8.7 SmNewRsp StateMod Response

The StateMod Response File preprocessor (SmNewRsp) allows a discontinued StateMod response file to be read and a new StateMod file constructed that is consistent with the Version 10.30 update that allows files to be provide in any order using a file descriptor. As presented below SmNewRsp keys on the suffix recommended and typically used in an existing StateMod response file to build a control file that is consistent with version 10.30 and greater. Note that if any existing file that does not contain a standard, recommended name SmNewRsp will warn the user but will not try to determine the file type. Also, since the new response file format was adopted with version 10.30, any files added after that time (e.g. Plans, Reservoir Return Data, etc.) are not processed. Similar to files with a non-standard suffix, SmNewRsp will warn the user but will not try to determine the file type. When a warning is encountered, the output from SmNewRsp will typically require hand editing before they can be successfully used by StateMod.

To execute SmNewName the user simply types:

SmNewName ffname.rsp

where ffname.rsp is an old sequential StateMod response. The new random response file is named SmNewRsp.out. Also execution notes and warnings are reported in file named SmNewRsp.log.

**Files processed by SmNewRsp (those in existence before version 10.30)**

#	Standard Suffix	File Descriptor	Example Name
1	*.ctl	Control	rgTWD.ctl
2	*.rin	River Network	rgTW.rin
3	*.res	Reservoir Station	rgTW.res
4	*.dds	Diversion Station	rgTW.dds
5	*.ris	StreamGage Station	rgTW.ris
6	*.ifs	Instreamflow Station	rgTW.ifs
7	*.wes	Well Station	rgTW.wes
8	*.ifr	Instreamflow Right	rgTW.ifr
9	*.rer	Reservoir Right	rgTW.rer
10	*.ddr	Diversion Right	rgTW.ddr
11	*.opr	Operational Right	rgTW.opr
12	*.wer	Well Right	rgTW.wer
13	*.dum	Precipitation Monthly	rgTW.pre
14	*.eva	Evaporation Monthly	rgTW.eva
15	*.rim	Stream Base Monthly	rgtw.rim
16	*.ddm	Diversion Demand Monthly	rgTW.ddm
17	*.dda	Diversion Demand Average Monthly	rgTW.dda
18	*.ddo	Diversion Demand Override	rgTW.ddo
19	*.ifm	Instreamflow Demand Monthly	rgTW.ifm
20	*.ifa	Instreamflow Demand AverageMonthly	rgTW.ifa
21	*.wem	Well Demand Monthly	rgTW.wem
22	*.dly	DelayTable Monthly	rgTW.dly
23	*.tar	Reservoir Target Monthly	rgTW.tar
24	*.ipy	IrrigationPractice Yearly	rg.ipy
25	*.iwr	ConsumptiveWaterRequirement Monthly	rg.iwr
26	*.par	SoilMoisture	rg.par
27	*.eom	Reservoir Historic Monthly	rgTW.eom
28	*.rib	StreamEstimate Coefficients	rgTW.rib
29	*.rih	StreamGage Historic Monthly	rgTW.rih
30	*.ddh	Diversion Historic Monthly	rgTW.ddh
31	*.weh	Well Historic Monthly	rgTW.weh
32	*.gvp	GeographicInformation	rgTW_StateMod.gvp
33	*.out	OutputRequest	RgTW.out
34	*.rid	Stream Base Daily	rgTWD.rid
35	*.dum	Dummy	rgTWD.dum
36	*.dum	Dummy	rgTWD.dum
37	*.dum	Dummy	rgTWD.dum
38	*.dum	Dummy	rgTWD.dum
39	*.dld	DelayTable Daily	rgTwD.dld
40	*.iwd	ConsumptiveWaterRequirement Daily	rgTWD.iwd
41	*.rhy	StreamGage Historic Daily	rgTWD.rhy
42	*.dhy	Diversion Historic Daily	RgTWD.dhy
43	*.why	Well Historic Daily	RgTWD.why
44	*.eoy	Reservoir Historic Daily	RgTWD.eoy

## 8.8 SmDelay StateMod Delay File Program

The StateMod Delay File Program (SmDelay) allows a daily StateMod delay file (\*.dly or \*.urD) to be created from an existing StateMod monthly delay file (\*.dly or .urM). The approach used to estimate daily data is to construct a pattern by connecting the midpoints of monthly data. The result is a smooth daily estimate.

To execute SmDelay the user simply types:

```
SmDelay fname.rsp
```

where fname.rsp is a response file that includes the name of the existing monthly delay file and the name of the new daily delay file to be created. Following is an example:

```
#
# smDelay.rsp;
# Response file to create a daily delay file from a monthly file
#
#
# Name
#-----
rg2005.urM           Existing Monthly delay file
rg2005.urD           New Daily delay file
```