LimnoTech
Water | Environment | Scientists | Engineers

MEMORANDUM

FROM: Tad Slawecki

TO: Files

DATE: 10/31/2011

PROJECT: CALGUI

SUBJECT: CalLite 2.0 Graphical User Interface – Users' Documentation

This CalLite 2.0 GUI User's Documentation provides necessary information for model developers to understand and modify the highly reconfigurable graphical user interface (GUI) for the CalLite 2.0 water resource management model developed jointly by the U.S. Bureau of Reclamation (Reclamation) and the California Division of Water Resources (DWR). It is written as a reference for model developers who need to extend GUI capabilities to allow CalLite 2.0 users to provide inputs necessary to characterize and control features added to the CalLite 2.0 model implementation in WRIMS. CalLite 2.0 users should consult the CalLite 2.0 User's Guide provided in the download for details about how to run the CalLite 2.0 model.

The CalLite 2.0 GUI User's Documentation is divided into the following sections:

- CalLite 2.0 Architecture
- Defining the UI: GUI.xml
- Connecting the UI to model inputs
- Customized reporting options in the UI
- Coding and implementation notes

CalLite 2.0 Architecture

As shown in Figure 1, the architecture of the overall CalLite 2.0 modeling system as implemented is basically identical to the architecture originally presented in specification documents prepared by Reclamation and DWR. The only differences are (1) scenario information is currently stored in a text file with the extension ".cls" instead of in a DSS file and (2) GUI configuration information is stored in several separate files rather than in a single file named "GUI.xml".

The contents of the separate files are discussed below.

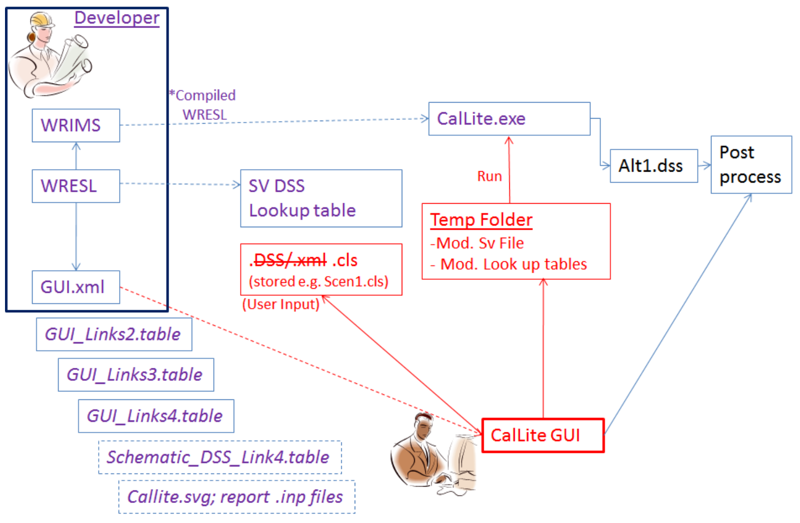

Developer
*Compiled WRESL
WRIMS
WRESL
GUI.xml
GUI_Links2.table
GUI_Links3.table
GUI_Links4.table
Schematic_DSS_Link4.table
Callite.svg; report .inp files
CalLite.exe
SV DSS Lookup table
~~.DSS/.xml~~ .cls (stored e.g. Scen1.cls) (User Input)
Run
Temp Folder -Mod. Sv File - Mod. Look up tables
Alt1.dss
Post process
CalLite GUI

Figure 1. CalLite 2.0 system architecture.

501 Avis Drive
Ann Arbor, MI 48108
734-332-1200
Fax: 734-332-1212
www.limno.com

Defining the UI: GUI.xml

The basic layout and content of the CalLite 2.0 GUI are controlled by an Extensible Markup Language (XML) file named GUI.xml. This file uses the SwiX^{ml} 2.0 specification to describe the Java Swing elements making up the GUI. Inside GUI.xml, controls are defined with XML tags that control organization and inputs. The following tags are heavily used in GUI.xml:

For organization and layouts:

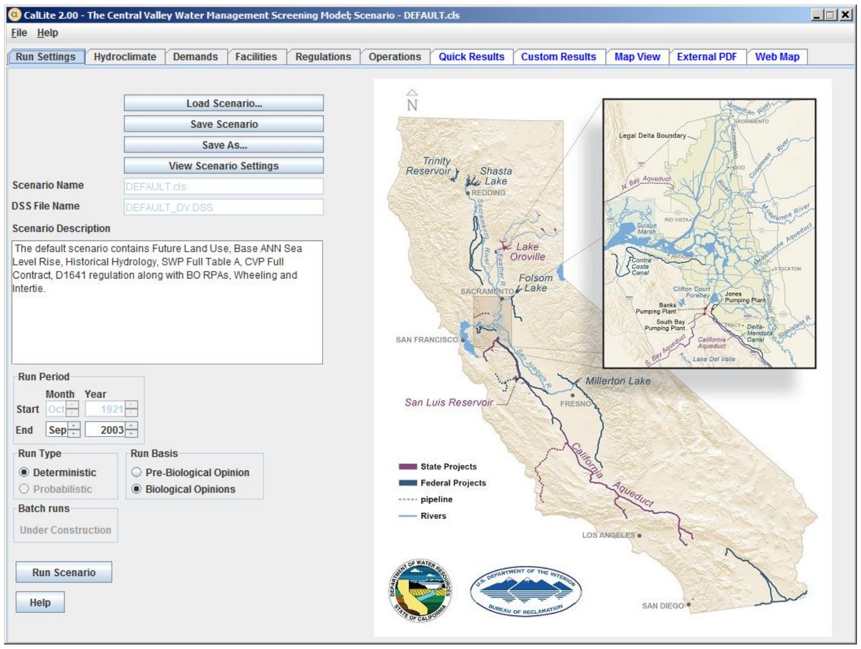
- <frame> - a top-level window
- <menubar> - a list of user-selectable actions displayed at the top of frame
- <panel> - an area within a window for display of information.
- <tabbedpane> - allows multiple panels to share the same space in a window
- <label text="text"/> - a text string
- <label icon="image"/> - a picture
- <gridbagconstraints> information that details where elements are displayed within a frame or panel

For input:

- <checkbox> - a simple on/off selection mechanism
- <buttongroup> - as shown here, used with radiobuttons to allow multiple-choice selection
    <radiobutton />
    <radiobutton />
  </buttongroup>
- <textfield> or <numtextfield> - input of text or numeric values.

Some additional input types (e.g. spinners and tables) are handled separately within the GUI code.

Figures 2 and 3 illustrates how <frame>, <menubar>, <panel> and <tabbedpane> elements are used to organize the layout of the CalLite 2.0 GUI. The menu and the tab-accessible panels in Figure 2's GUI snapshot are specified by XML codes shown in Figure 3. The XML code is shown in an outline view inside the Eclipse editor; additional detail that specifies the controls on individual panels is hidden in this view.

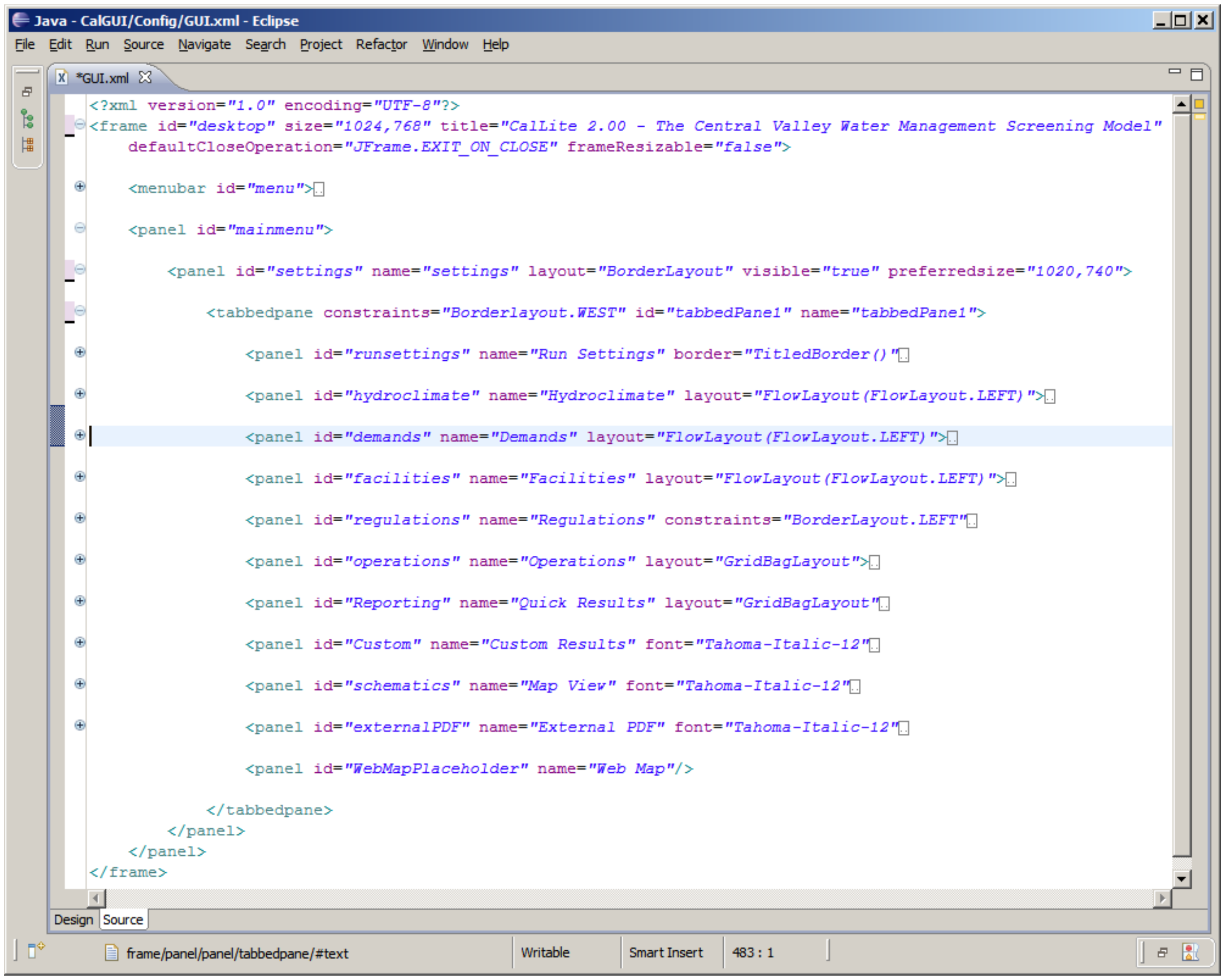


Scenario Name: DEFAULT.cls
DSS File Name: DEFAULT_DV.DSS
Scenario Description: The default scenario contains Future Land Use, Base ANN Sea Level Rise, Historical Hydrology, SWP Full Table A, CVP Full Contract, D1641 regulation along with BO RPAs, Wheeling and Intertie.

Run Period
Month Year
Start Oct 1921
End Sep 2003

Run Type: [✓ Deterministic] Probabilistic
Run Basis: Pre-Biological Opinion [✓ Biological Opinions]
Batch runs: Under Construction

Figure 2. Snapshot of main dashboard for CalLite 2.0 GUI.

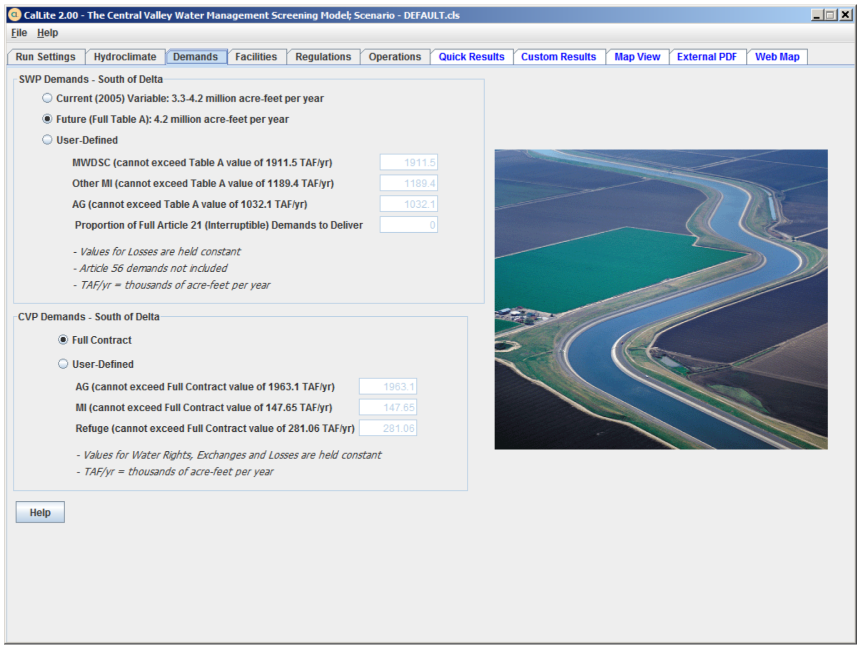

Java - CalGUI/Config/GUI.xml - Eclipse
File Edit Run Source Navigate Search Project Refactor Window Help
*GUI.xml
Design Source

```
<?xml version="1.0" encoding="UTF-8"?>
<frame id="desktop" size="1024,768" title="CalLite 2.00 - The Central Valley Water Management Screening Model"
    defaultCloseOperation="JFrame.EXIT_ON_CLOSE" frameResizable="false">

    <menubar id="menu">

    <panel id="mainmenu">

        <panel id="settings" name="settings" layout="BorderLayout" visible="true" preferredsize="1020,740">

            <tabbedpane constraints="Borderlayout.WEST" id="tabbedPane1" name="tabbedPane1">

                <panel id="runsettings" name="Run Settings" border="TitledBorder()">

                <panel id="hydroclimate" name="Hydroclimate" layout="FlowLayout(FlowLayout.LEFT)">

                <panel id="demands" name="Demands" layout="FlowLayout(FlowLayout.LEFT)">

                <panel id="facilities" name="Facilities" layout="FlowLayout(FlowLayout.LEFT)">

                <panel id="regulations" name="Regulations" constraints="BorderLayout.LEFT">

                <panel id="operations" name="Operations" layout="GridBagLayout">

                <panel id="Reporting" name="Quick Results" layout="GridBagLayout">

                <panel id="Custom" name="Custom Results" font="Tahoma-Italic-12">

                <panel id="schematics" name="Map View" font="Tahoma-Italic-12">

                <panel id="externalPDF" name="External PDF" font="Tahoma-Italic-12">

                <panel id="WebMapPlaceholder" name="Web Map"/>

            </tabbedpane>
        </panel>
    </panel>
</frame>
```

```
frame/panel/panel/tabbedpane/#text    Writable    Smart Insert    483 : 1
```

Figure 3. GUI.xml excerpt (in outline view) showing GUI components defining main CalLite 2.0

The GUI Demands dashboard in Figure 4 is generated by the XML shown in Figure 5, which is an excerpt of the GUI.xml file in detailed view that includes input control elements such as <radiobutton> and <numtextfield>.



SWP Demands - South of Delta

- ◯ Current (2005) Variable: 3.3-4.2 million acre-feet per year
- ● Future (Full Table A): 4.2 million acre-feet per year
- ◯ User-Defined

MWDSC (cannot exceed Table A value of 1911.5 TAF/yr)	1911.5
Other MI (cannot exceed Table A value of 1189.4 TAF/yr)	1189.4
AG (cannot exceed Table A value of 1032.1 TAF/yr)	1032.1
Proportion of Full Article 21 (Interruptible) Demands to Deliver	0

- Values for Losses are held constant
- Article 56 demands not included
- TAF/yr = thousands of acre-feet per year

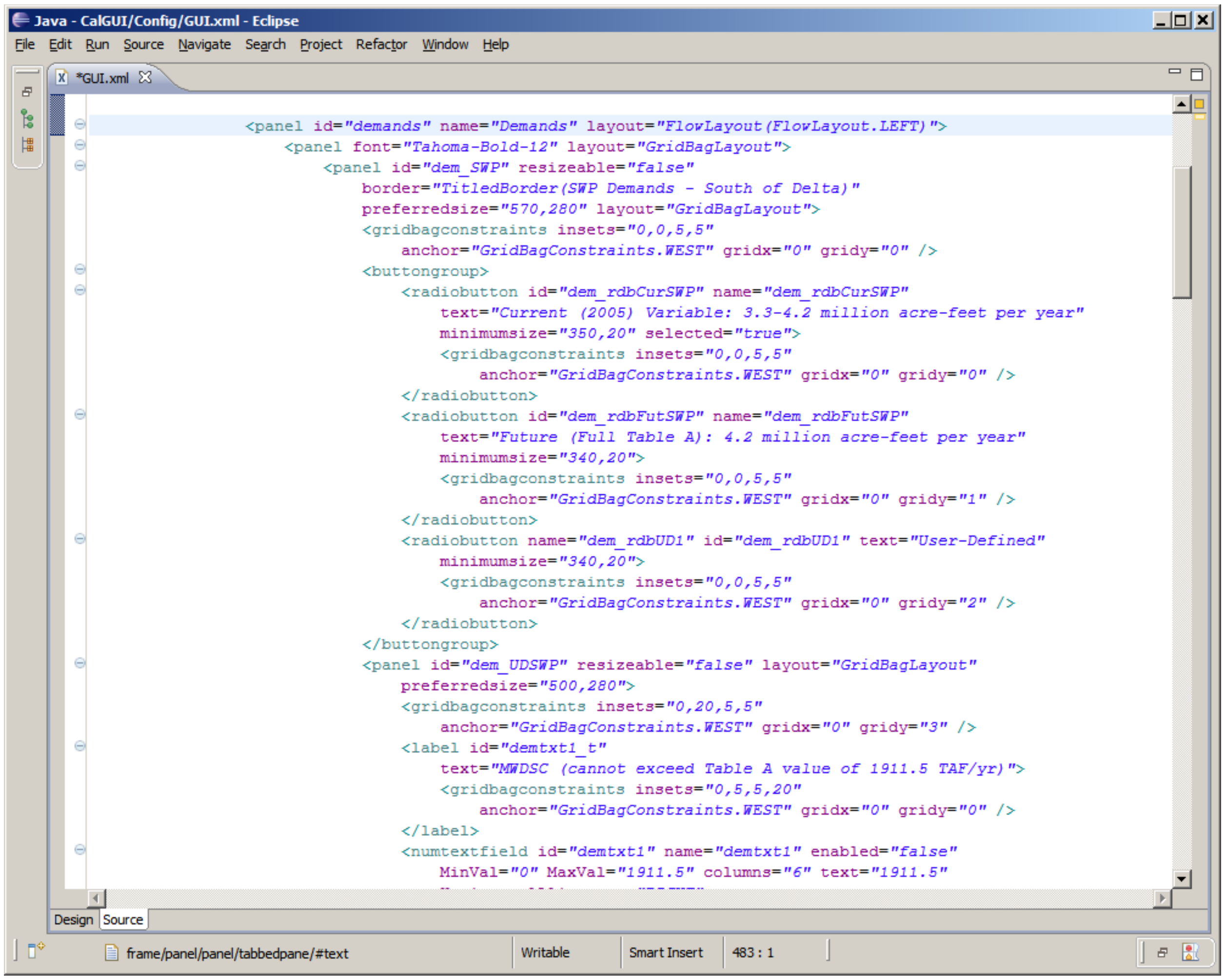
CVP Demands - South of Delta

- ● Full Contract
- ◯ User-Defined

AG (cannot exceed Full Contract value of 1963.1 TAF/yr)	1963.1
MI (cannot exceed Full Contract value of 147.65 TAF/yr)	147.65
Refuge (cannot exceed Full Contract value of 281.06 TAF/yr)	281.06

- Values for Water Rights, Exchanges and Losses are held constant
- TAF/yr = thousands of acre-feet per year

Figure 4. Snapshot of demand dashboard for CalLite 2.0 GUI.


Java - CalGUI/Config/GUI.xml - Eclipse
File Edit Run Source Navigate Search Project Refactor Window Help
*GUI.xml
Design Source
frame/panel/panel/tabbedpane/#text
Writable
Smart Insert
483 : 1

```
<panel id="demands" name="Demands" layout="FlowLayout(FlowLayout.LEFT)">
    <panel font="Tahoma-Bold-12" layout="GridBagLayout">
        <panel id="dem_SWP" resizeable="false"
            border="TitledBorder(SWP Demands - South of Delta)"
            preferredsize="570,280" layout="GridBagLayout">
            <gridbagconstraints insets="0,0,5,5"
                anchor="GridBagConstraints.WEST" gridx="0" gridy="0" />
            <buttongroup>
                <radiobutton id="dem_rdbCurSWP" name="dem_rdbCurSWP"
                    text="Current (2005) Variable: 3.3-4.2 million acre-feet per year"
                    minimumsize="350,20" selected="true">
                    <gridbagconstraints insets="0,0,5,5"
                        anchor="GridBagConstraints.WEST" gridx="0" gridy="0" />
                </radiobutton>
                <radiobutton id="dem_rdbFutSWP" name="dem_rdbFutSWP"
                    text="Future (Full Table A): 4.2 million acre-feet per year"
                    minimumsize="340,20">
                    <gridbagconstraints insets="0,0,5,5"
                        anchor="GridBagConstraints.WEST" gridx="0" gridy="1" />
                </radiobutton>
                <radiobutton name="dem_rdbUD1" id="dem_rdbUD1" text="User-Defined"
                    minimumsize="340,20">
                    <gridbagconstraints insets="0,0,5,5"
                        anchor="GridBagConstraints.WEST" gridx="0" gridy="2" />
                </radiobutton>
            </buttongroup>
            <panel id="dem_UDSWP" resizeable="false" layout="GridBagLayout"
                preferredsize="500,280">
                <gridbagconstraints insets="0,20,5,5"
                    anchor="GridBagConstraints.WEST" gridx="0" gridy="3" />
                <label id="demtxt1_t"
                    text="MWDSC (cannot exceed Table A value of 1911.5 TAF/yr)">
                    <gridbagconstraints insets="0,5,5,20"
                        anchor="GridBagConstraints.WEST" gridx="0" gridy="0" />
                </label>
                <numtextfield id="demtxt1" name="demtxt1" enabled="false"
                    MinVal="0" MaxVal="1911.5" columns="6" text="1911.5"
```

Figure 5. GUI.xml excerpt (in detailed view) showing GUI components defining demand dashboard.

When editing GUI.xml, **<u>it is extremely important</u>** to assign "id" and "name" attributes to each element that corresponds to a model input:

<radiobutton **id="dem_rdbCurSWP" name="dem_rdbCurSWP"**/>

The name and id attributes should be set to the same value. This value is used to link the control element in the GUI to model inputs that the CalLite 2.0 model reads from model lookup tables.

Connecting the UI to model inputs

The name and id attributes are used interchangeably to connect control elements in the CalLite 2.0 GUI with model inputs that are written to .table files used by the CalLite 2.0 model. The configuration files GUI_Links2.table and GUI_Links4.table are used to provide this connection.

As shown in Figure 6, the GUI_Links2.table file (tab-delimited text) provides a lookup table name and location for selected input controls.

GUI id	Table name	Index	Option	Description	Dashboard	Data tables	SwitchID	TableID
dem_rdbCurSWP	GUI_SODdemand.table	0	1	SWP demand type 1=non-user-defined demands	Demands			
dem_rdbFutSWP	GUI_SODdemand.table	0	1	SWP demand type 1=non-user-defined demands	Demands			
dem_rdbUD1	GUI_SODdemand.table	0	2	SWP demand type 2=user-defined demands	Demands			
ckbReg3	GUI_D1641Regs.table	1	-2	DCC_DLTSW - Delta Cross Channel	Regulations	gui_xchanneldays	3	1
ckbReg15	GUI_D1641Regs.table	6	-1	X2ROE_DLTSW - Roe Trigger	Regulations			

Figure 6. Excerpt from GUI_Links2.table file.

CalLite inputs are defined in individual lookup .table files corresponding to dashboards, and the information in GUI_Links2.table explains to the GUI how to write the lookup files. GUI_Links2.table also identifies which control(s) in the GUI determine the value to be written.

The following information is provided for each managed control:

- **GUI id**. The value for the name and id attributes of the SwiXml element in the GUI.xml file corresponding to the control.
- **Table name**. The name of the table where the value(s) of the control will be written.
- **Index**. The index (row number) in the table where the value(s) of the control will be written.
- **Option**. Specifies the value to be written for the control. If greater than zero, the control is a radiobutton, and the value specified is written only if that radiobutton has been selected. If the option value is -1, the value entered for the corresponding textfield or numtextfield control is written. If the value is -2, the table specified by TableID is written.
- **Description**. For non-tabular values, this string is also written to the file.
- **Dashboard**. Identifies the name of the GUI dashboard where the control is located.
- **Data Tables**. Name of the default data file for data tables; the GUI program will read initial values for the table from this data file.
- **SwitchID and TableID**. TableID is a unique identifier that sets the internal storage location in the GUI for table information. SwitchID is a separate identifier that fulfills a very similar, nearly redundant function. When adding a new table to CalLite 2.0, the TableID value can be set to the largest current TableID value plus one. If the new TableID value is greater than 10, the new SwitchID should be set to the same value; the pairings of SwitchID and TableID forvaluesless than or equal to 10 should not be changed. The code artifact that requires entry of both values (SwitchID and TableID) should be corrected in the next phase of CalLite GUI development, leaving only one of the two identifiers.

Figure 7 shows the contents of the GUI_Links4.table file, which identifies the SV and Init file name and F-Parts to be used along with the CVP_WSI_DI and SWP_WSI_DI tables.

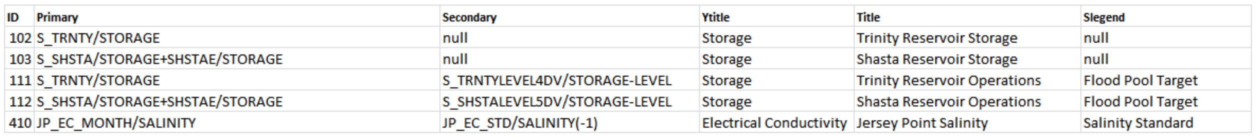
RunBasis_ID	LOD_ID	CCProject_ID	CCModel_ID	SV_File	F_PartSV	Init_File	F_PartSV	CVP_WSI_DI	SWP_WSI_DI
1	1	1	0	CL_EXISTING_PREBO_081011_SV.DSS	2005A01A	CL_INIT.dss	INITIAL	\WSI_DI\wsi_di_cvp_sys_1110.table	\WSI_DI\wsi_di_swp_1110.table
1	2	1	0	CL_FUTURE_PREBO_080911_SV.DSS	2020D09E	CL_INIT.dss	INITIAL	\WSI_DI\wsi_di_cvp_sys_1210.table	\WSI_DI\wsi_di_swp_1210.table
2	1	1	0	CL_EXISTING_BO_081011_SV.DSS	2005A01A	CL_INIT.dss	INITIAL	\WSI_DI\wsi_di_cvp_sys_2110.table	\WSI_DI\wsi_di_swp_2110.table
2	2	1	0	CL_FUTURE_BO_080911_SV.DSS	2020D09E	CL_INIT.dss	INITIAL	\WSI_DI\wsi_di_cvp_sys_2210.table	\WSI_DI\wsi_di_swp_2210.table

Figure 7. Contents of GUI_Links4.table

The GUI selects the files and F-Parts based on the values of the Run Basis, Level of Development, Climate Period Projections, and Climate Change Models options set by the user. Additional combinations may be added to the file.

Customized reporting options in the UI

The GUI_Links3.table and Schematic_DSS_Link4.table files are tab-delimited text files that specify output linkages and display/labeling options for different controls in the GUI.



ID	Primary	Secondary	Ytitle	Title	Slegend
102	S_TRNTY/STORAGE	null	Storage	Trinity Reservoir Storage	null
103	S_SHSTA/STORAGE+SHSTAE/STORAGE	null	Storage	Shasta Reservoir Storage	null
111	S_TRNTY/STORAGE	S_TRNTYLEVEL4DV/STORAGE-LEVEL	Storage	Trinity Reservoir Operations	Flood Pool Target
112	S_SHSTA/STORAGE+SHSTAE/STORAGE	S_SHSTALEVEL5DV/STORAGE-LEVEL	Storage	Shasta Reservoir Operations	Flood Pool Target
410	JP_EC_MONTH/SALINITY	JP_EC_STD/SALINITY(-1)	Electrical Conductivity	Jersey Point Salinity	Salinity Standard

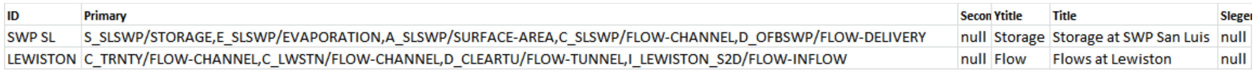
Figure 8. Excerpt from GUI_Links3.table

Shown in Figure 8, the GUI_Links3.table file identifies DSS data series to display when certain controls are double-clicked or checked in the Quick Results dashboard. The necessary fields are:

- **ID.** The value for the name and id attributes of the SwiXml element in the GUI.xml file corresponding to the checkbox control in the Quick Results dashboard.
- **Primary.** The B-Part/C-Part specification for the first data set to be included in a display. The first display set can also be the sum of several DSS data sets if the specifications are combined with a plus sign ("+").
- **Secondary.** The B-Part/C-Part specification for the (optional) second data set to be included in a display. A displayed data set can be shifted to the previous month by appending the string "(-1)" to the specification.
- **YTitle.** Y-axis title.
- **Title.** Chart/table title.
- **SLegend.** Legend text for secondary data set.

The experimental Schematic_DSS_Links4.table file establishes similar links between model time series and the Map View dashboard in the GUI, with the following differences:

1. The ID refers to the text value associated with an SVG element clicked on in the schematic view.
2. If the ID/text value contains an underscore – e.g. S_TRNTY or C_ALAMO – the remaining fields are treated as in GUI_Links3.table and generate a display similar to Quick Results.
3. If the ID/text value does not contain an underscore (e.g. LEWISTON) and the Primary field does not include any EVAPORATION time series, all of the comma-separated data sets will be plotted on a single chart.
4. If the ID/text value does not contain an underscore and the Primary field includes an EVAPORATION time series, the first series listed is put in a time series chart labeled STORAGE, the next two in a dual-axis chart for surface area and evaporation, and the last one or two series in a third chart labeled Flow, all displayed on the same chart panel.
5. If the ID/text value does not match with case 2, 3 or 4 above, the currently selected display options will be applied to the first time series listed for the active scenario only.



ID	Primary	Secon	Ytitle	Title	Slege
SWP SL	S_SLSWP/STORAGE,E_SLSWP/EVAPORATION,A_SLSWP/SURFACE-AREA,C_SLSWP/FLOW-CHANNEL,D_OFBSWP/FLOW-DELIVERY	null	Storage	Storage at SWP San Luis	null
LEWISTON	C_TRNTY/FLOW-CHANNEL,C_LWSTN/FLOW-CHANNEL,D_CLEARTU/FLOW-TUNNEL,I_LEWISTON_S2D/FLOW-INFLOW	null	Flow	Flows at Lewiston	null

Figure 9. Example Schematic_DSS_Links4.table records for multi-chart and multiple-series displays

Figure 9 shows example records from the Schematic_DSS_Links4.table file that creates multiple-chart and multiple-series displays.

The schematic view referenced by the Schematic_DSS_Links4.table file is drawn from the file callite.svg, a ScalableVector Graphics (SVG) file that presents a simplified depiction of the model domain. The GUI looks for the first text content associated with a clicked object in the SVG file, so each schematic element with an output context in the SVG file must be grouped with a text element that is included in theSchematic_DSS_Links4.table file.

Additional customizable output options are incorporated in the GUI but supported by DWR:

- Browser-based map-centric view of system. DWR has implemented a Google Maps-based web page that can be instanced within the GUI. Clicking on items within the map returns an ID that can be interpreted in the same manner as Quick Result views using GUI_Links3.table.
- Automated report tool. DWR's existing PDF report generation tool is incorporated into the GUI through a customized dashboard that allows the user to select a report template and enter necessary key fields. The initial distribution of CalLite 2.0 includes the templates callite_scenario_comparison.inp and calsim_callite_corroboration.inp. See Appendix I of the CalLite 2.0 Reference Manual for description of these templates.

Coding and implementation notes

This section consists of various notes related to the implementation of the GUI that may be of use to CalLite model developers needing a more detailed understanding of the GUI's inner workings.

Code site: The CalLite GUI Development site is at http://code.google.com/p/callitegui/.

Compiler: The CalLite 2.0 GUI was developed using Java SE 6 with Eclipse Helios SR1.

Directory structure: all GUI configuration files (GUI.xml, GUI_Links*.table, Schematic_DSS_Links4.table, callite.svg, and 8.inp report templates) are stored in the

```
.\                      (CalLiteGUI.jar, CalLite.bat)
 .\Config               (GUI.xml, GUI_Links, templates)
 .\Default              (study.sty, LF90.EER)
  .\Default\DSS
  .\Default\External
 .\Default\Lookup
 .\Model                (main_BO.exe, DLLs)
 .\Run
  .\Run\DSS
  .\Run\External
  .\Run\Lookup
 .\Scenarios            (CalLite scenarios, results)
```

Libraries: The CalLite 2.0 GUI references the following major libraries:

- batik – java SVG toolkit
- callite_report – DWR library for template-driven report generation
- javaHeclib – java toolkit for access to HEC DSS files
- jhelp – java toolkit for help files
- jbrowser – Embedded browser support
- jfreechart – charting objects for java.
- vista – DWR legacy library for DSS handling

Web resources:

- XML - http://www.w3schools.com/web/web_xml.asp
- SwiX^{ml} - http://www.swixml.org/
- Swing (Java user interface library) - http://docs.oracle.com/javase/tutorial/uiswing/
- gridbaglayout (layout manager) -
  http://docs.oracle.com/javase/tutorial/uiswing/layout/gridbag.html

XML Hints:

- XML needs to be properly nested and closed

  - INCORRECT:
    <buttongroup name="bg1">
    <radiobutton name="rdb1" id="rdb1" text="Opt1">
    </buttongroup>
    </radiobutton>

  - CORRECT:
    <buttongroup name="bg1">
    <radiobutton name="rdb1" id="rdb1" text="Opt1">
    </radiobutton>
    </buttongroup>

- Certain attributes are useful for display:

  - Text="text to display"; this is also the value of a textfield
  - Selected="true"/"false" – initial state for checkboxes and radiobuttons
  - Enabled="true"/"false" for display without action
  - The id and name attributes must be specified for controls referenced in GUI_Links2.table.