RiverWare Model Report Generator

High Level Design

Phil Weinstein, CADSWES

This document provides a high-level design for a configurable RiverWare model report generator. This is a general tool, but supports the specific requirements defined by this document:

- USACE Text Output: Functional Requirements (Draft, 2-12-2009, 8 pages)
  Authors: David Neumann and Edie Zagona (CADSWES), and John Daylor (USACE-Tulsa)

The user can select and specify the order of report sections and model objects, including Subbasins, Simulation Objects, Accounts and Slots. Output for large Series and Table Slots can be truncated according to specified timestep or row count limits. Both plain text and HTML output generation is supported. The user can supply CSS (cascading style sheets) text styles for the HTML output.

Special feature: The output of Reservoir Table Slots having elevation, storage volume, or surface area values can be augmented with equivalent entities based on the Reservoir's Elevation Volume Table Slot and Elevation Area Table Slot. There is an option to express storage volumes also as percentages of Conservation Pool and Flood Pool levels.

0.1 Other Related Documents

RiverWare Model Report Generation Demo (3-31-2009, minor revision 4-3-2009, 14 pages).
Author: Phil Weinstein (CADSWES).

0.2 Document Status

03-08-2009: Ready for review.
04-02-2009: Revisions after review.
04-08-2009: Additional revisions -- Ready for review.

- Present simpler alternative to management of Report Configurations as "Output Devices".
- Remove two custom object-level orderings defined in the report configuration. The user will be able to *choose* the Subbasins' custom SimObj orderings, and the SimObjs' custom Slot orderings, as options, but will not be able to *define* new custom orderings for those entities in the report configuration.
- Added section on Report Configuration Persistence (Serialization) implementation options.
- Added analysis of the effect on RiverWare Model Size of storing Report Configurations within the model.
- Revised development estimates.

0.3 Contents Summary

0.4 Contents

1.0 Report Configuration and Generation Process

The format and content specification of a generated model report is represented by a report configuration object. Report configurations are composed and edited using the RiverWare Report Configuration dialog box; *see the next section.*

We are considering *two design options* for the management of report configuration objects, described in the following subsections. Depending on the approach we take, the RiverWare Report Configuration dialog box is shown either through the Output Manager dialog, or through the RiverWare Workspace File menu.

The generation of a report always results in the creation of an external report file, either in plain text or HTML. An external application to view the report (generally a text editor or web browser) can be launched from RiverWare using some of the facilities of the RiverWare Link to External RPL Documents feature (RiverWare 5.1). This includes use of the File Type Association Manager to associate the filepath of an executable program with the TEXT and HTML file types, for the purpose of opening (for viewing or editing) files of those types.

The path and name of the generated report file is part of the report configuration. (See the “Output” Tab within the RiverWare Model Summary Report Configuration dialog box).

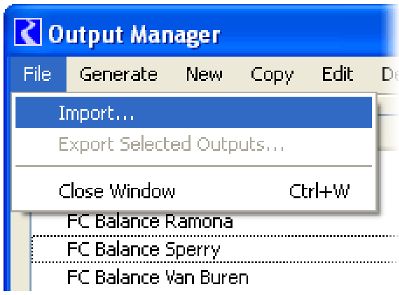
1.1 Option 1: Report Configurations as “Output Devices” within a RiverWare Model

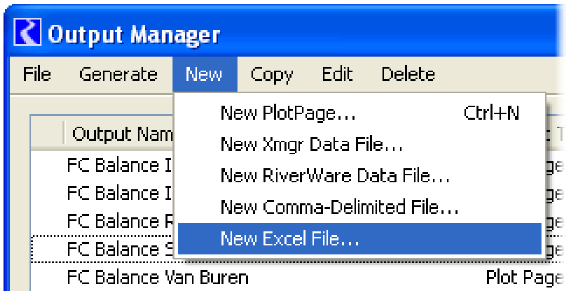
Report configurations are managed as “Output Devices” through the RiverWare Output Manager (which is accessible from the RiverWare Workspace Utilities menu). Multiple report configurations can be defined within a RiverWare model. They can be exported as report configuration files which can be imported to other RiverWare models via the Output Manager.

From the RiverWare Report Output Manager, the user can create a New Report, or Open an Existing Report. Both of those operations show the RiverWare Model Report Generator dialog box. From there, the user can:

- define and modify the report configuration,
- generate and open (in an external browser application) a report from the configuration,
- open a previously generated report (i.e. the one at the filepath specified by the configuration’s output path value), or
- export the report configuration to a report configuration file.

Management of Report Configurations as “Output Devices” will require some internal “Output Device” class development, and additional operations for “Model Reports” in the Generate, New, Copy, Edit and Delete menus of the Output Manager, partially illustrated here *(not including the additional items for Model Reports):*





1.2 Option 2: Independent Report Configuration Files

Report configurations are managed as distinct entities, not associated with a particular RiverWare model file. They can be independently loaded (from either the Workspace File menu or the Model Report Configuration dialog), and saved as RiverWare Report Configuration Files.

From the RiverWare Summary Report Configuration dialog box, the user can:

- start defining a new report configuration,
- load a previously saved configuration from a report configuration file,
- generate and view a report from the configuration,
- view a previously generated report (i.e. the one at the filepath specified by the configuration's output path value), or
- save the report configuration to a report configuration file.

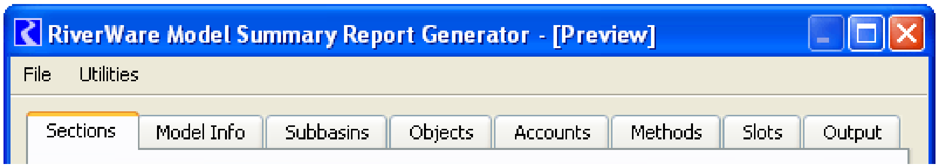
2.0 RiverWare Model Report Configuration Dialog

Report configurations are composed and edited using the RiverWare Report Configuration dialog box.

This dialog box is organized in eight tabs for the purpose of defining the report format and content. Report generation and opening (by launching an external viewer program) is also operated from this dialog box.

The configuration dialog box tabs are:

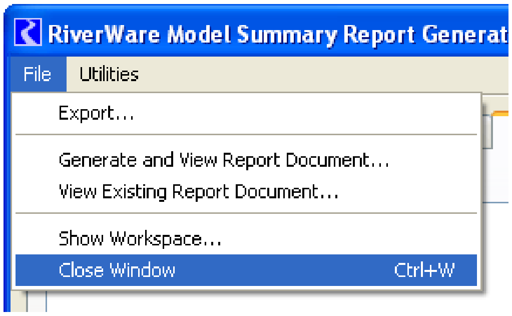
- Sections
- Model Info
- Subbasins
- Objects
- Accounts
- Methods
- Slots
- Output



2.1 Config Dialog: Menubar

The RiverWare Report Configuration dialog box has a menubar which includes “File” and “Utilities” menus. The composition of the File menu depends on the option chosen for the management of Report Configuration instances:

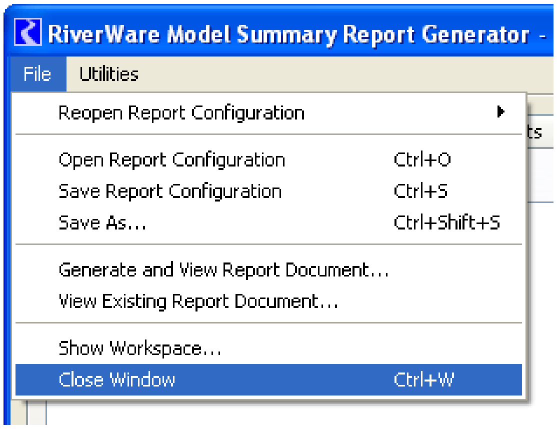
Option 1: Report Configurations as “Output Devices” within a RiverWare Model:


RiverWare Model Summary Report Generat
File
Utilities
Export...
Generate and View Report Document...
View Existing Report Document...
Show Workspace...
Close Window
Ctrl+W

The “Export” item open up a file selector, initialized to the loaded model directory. The exported file can be imported (generally into a different RiverWare model) from the Output Manager.

The “Generate and View Report Document...” and “View Existing Report Document...” items perform the same functions as similarly named push buttons, described below.

Option 2: Independent Report Configuration Files:

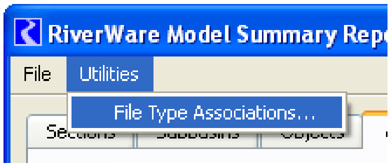

RiverWare Model Summary Report Generator -
File
Utilities
Reopen Report Configuration
Open Report Configuration
Ctrl+O
Save Report Configuration
Ctrl+S
Save As...
Ctrl+Shift+S
Generate and View Report Document...
View Existing Report Document...
Show Workspace...
Close Window
Ctrl+W

The “Reopen Report Configuration” item shows a submenu of recently saved or loaded config files and recently accessed directories. The directory items open up a file selector, initialized to the indicated directory.

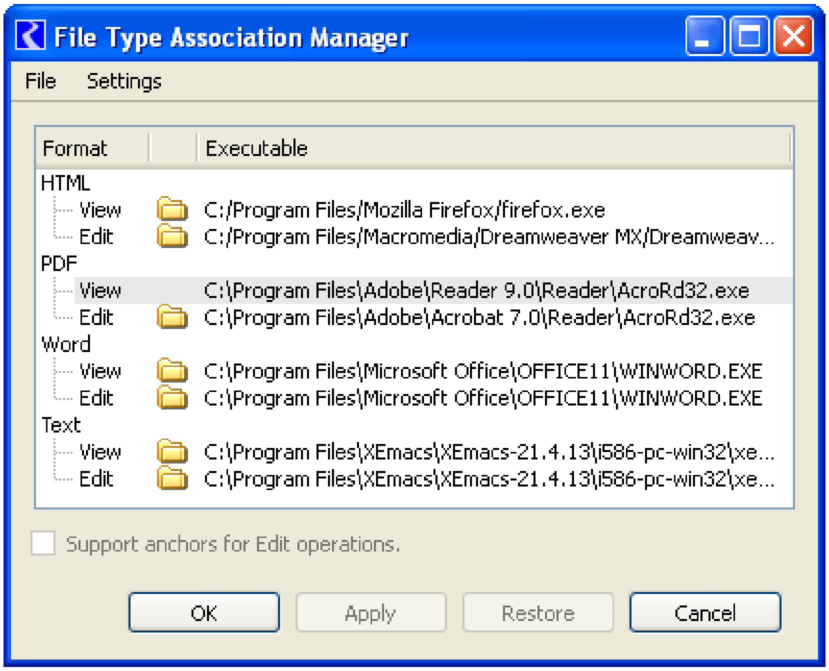
The “Open Report Configuration...” and “Save As...” menu items also open up a file selector.

Utilities Menu:

The “File Type Associations...” menu item opens up the File Type Association Manager dialog. This was introduced in RiverWare 5.1 as part of the RiverWare Link to External RPL Documents feature to allow the configuration within RiverWare of executables (programs) to be used for editing and viewing documents of various formats.



The File Type Associations Manager dialog is illustrated below.



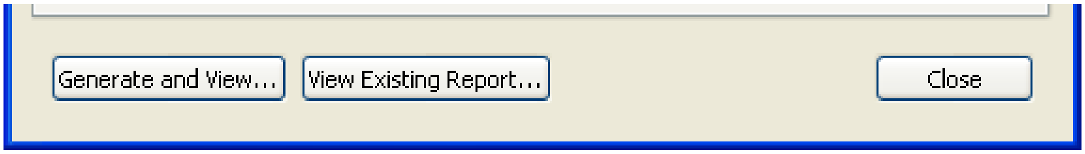
Format	Executable
HTML	
View	C:/Program Files/Mozilla Firefox/firefox.exe
Edit	C:/Program Files/Macromedia/Dreamweaver MX/Dreamweav...
PDF	
View	C:\Program Files\Adobe\Reader 9.0\Reader\AcroRd32.exe
Edit	C:\Program Files\Adobe\Acrobat 7.0\Reader\AcroRd32.exe
Word	
View	C:\Program Files\Microsoft Office\OFFICE11\WINWORD.EXE
Edit	C:\Program Files\Microsoft Office\OFFICE11\WINWORD.EXE
Text	
View	C:\Program Files\XEmacs\XEmacs-21.4.13\i586-pc-win32\xe...
Edit	C:\Program Files\XEmacs\XEmacs-21.4.13\i586-pc-win32\xe...

☐ Support anchors for Edit operations.

2.2 Config Dialog: Bottom Buttons

The RiverWare Report Configuration dialog box has the following three buttons along the bottom:

- Generate and View...
- View Existing Report...
- Close

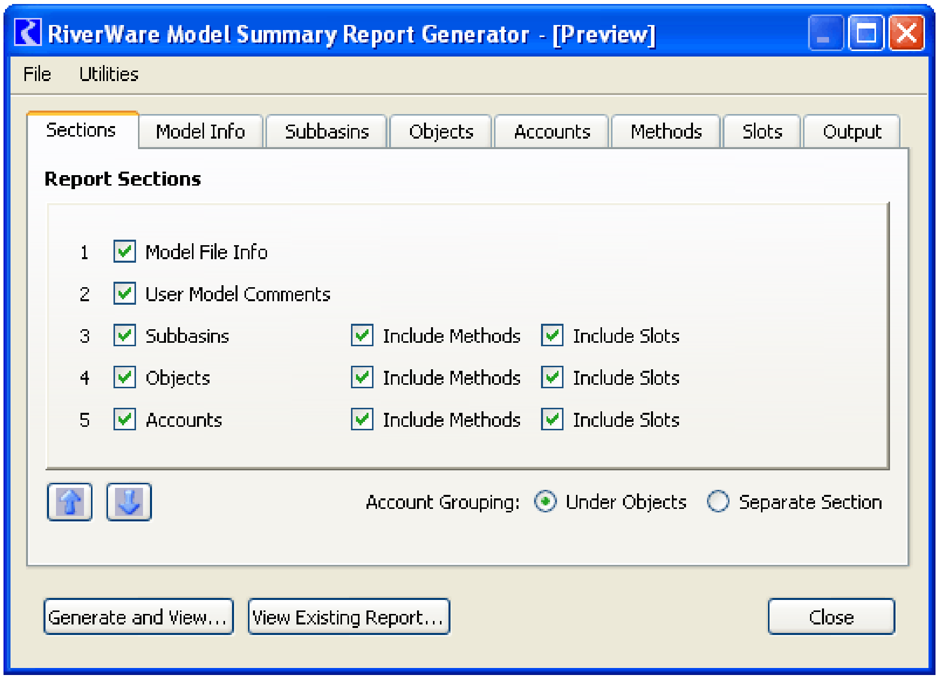


2.3 Config Dialog Tab 1: Sections

The Sections Tab allows the user to select which major sections appear in the generated report, and to specify their order.

The blue arrows move the selected item up or down.

This mock-up image doesn't correctly illustrate the organization of data within a table widget, with selectable rows. The one or more selected rows would be moved with the blue arrows.



Report Sections

1 ☑ Model File Info

2 ☑ User Model Comments

3 ☑ Subbasins ☑ Include Methods ☑ Include Slots

4 ☑ Objects ☑ Include Methods ☑ Include Slots

5 ☑ Accounts ☑ Include Methods ☑ Include Slots

Account Grouping: ● Under Objects ○ Separate Section

There are two ways of organizing Accounts within the report.

- Account instance sections can be arranged under their respective Simulation Object sections. If Object Slots are also included in the report, the Slot instance sections would come before the Account instance sections.
- Account instance sections can be arranged together after all of the Object instance sections.

The generated report content of various major sections is described in the Report Generation chapter in this document. The major sections have additional configuration on subsequent tabs of this configuration dialog.

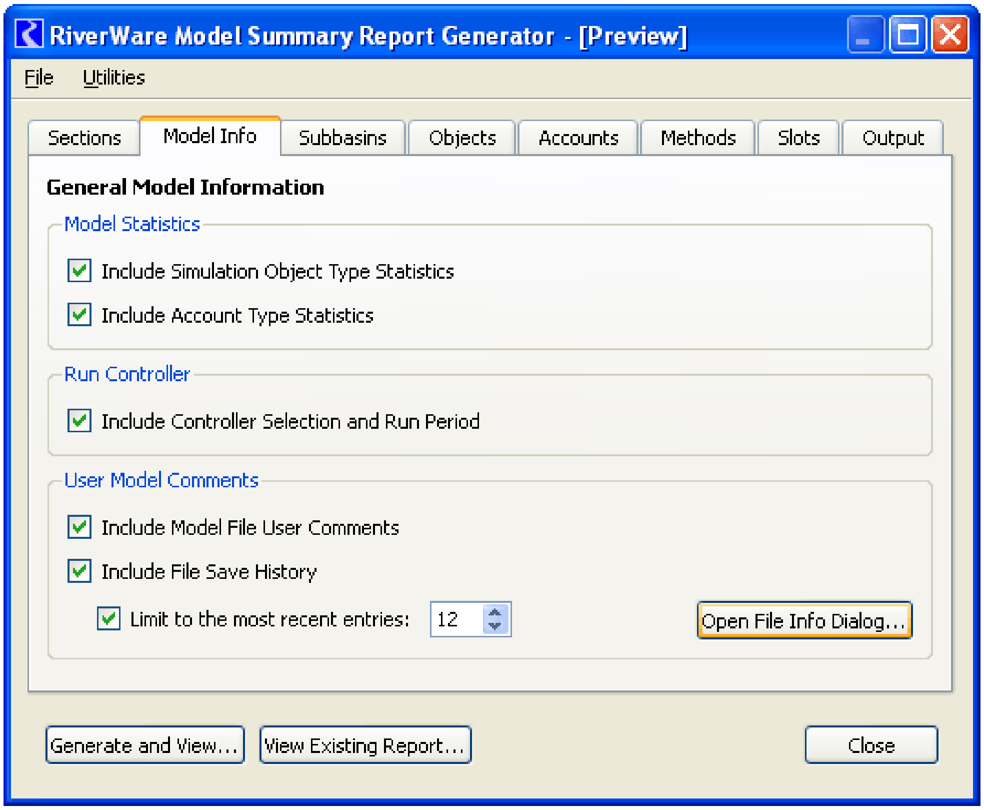
2.4 Config Dialog Tab 2: Model Info

The Model Info Tab controls the appearance of various general items in the report.

Clicking the “Open File Info Dialog...” button shows that dialog. *(See below).*

With that dialog, the user may revise the Model File User Comments and examine the File Save History items.

The number of “Save History” items written to the report can be limited to a specified number of most recent items.



General Model Information

Model Statistics

☑ Include Simulation Object Type Statistics

☑ Include Account Type Statistics

Run Controller

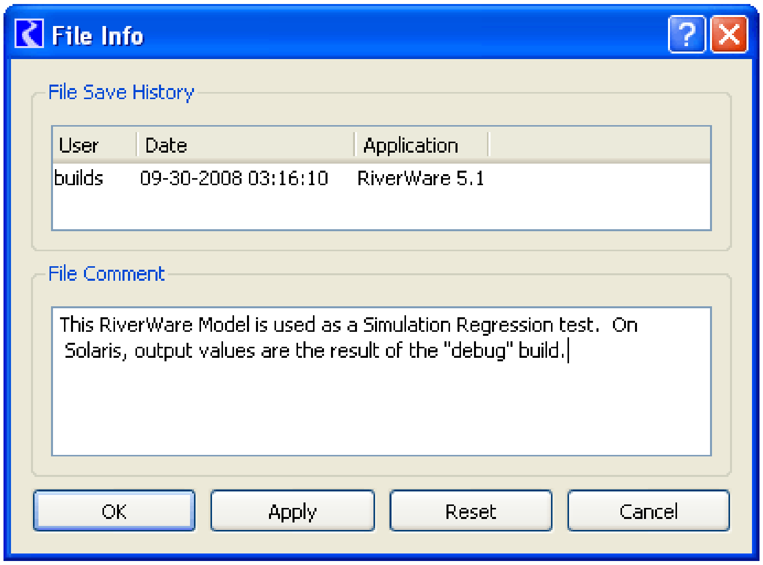
☑ Include Controller Selection and Run Period

User Model Comments

☑ Include Model File User Comments

☑ Include File Save History

☑ Limit to the most recent entries: 12



File Save History

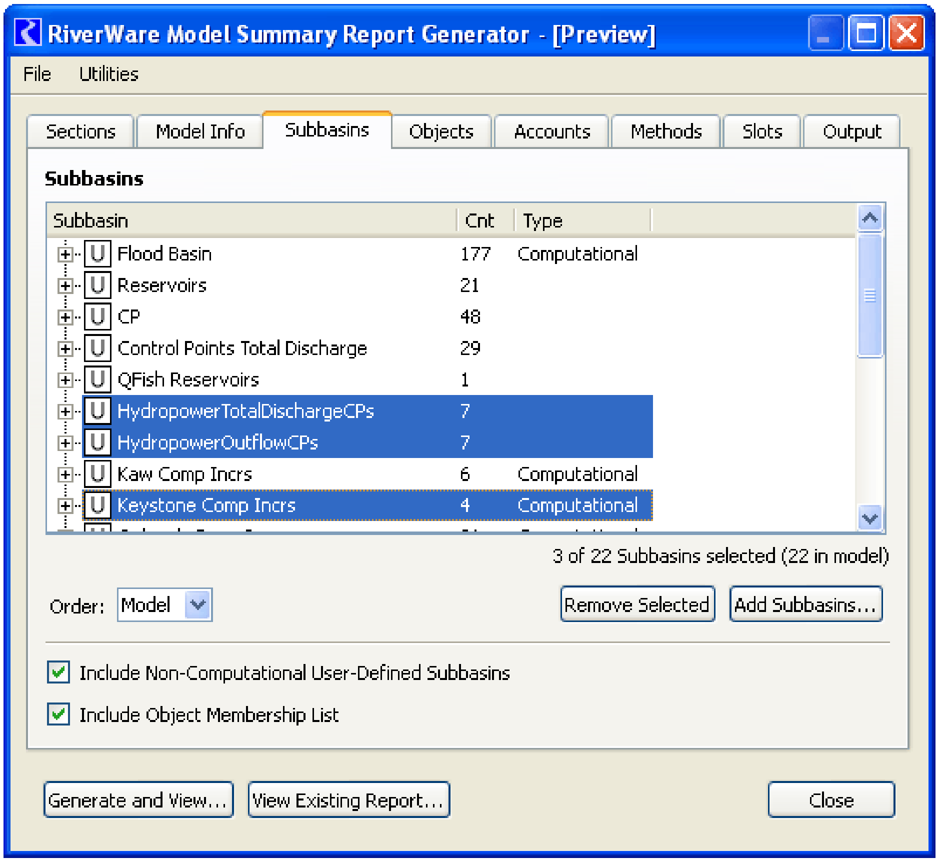
User	Date	Application
builds	09-30-2008 03:16:10	RiverWare 5.1

File Comment

This RiverWare Model is used as a Simulation Regression test. On Solaris, output values are the result of the "debug" build.

2.5 Config Dialog Tab 3: Subbasins

The Subbasins Tab indicates which, and in what order Subbasins appear in the subbasins section of the generated report document.



Subbasins

Subbasin	Cnt	Type
Flood Basin	177	Computational
Reservoirs	21	
CP	48	
Control Points Total Discharge	29	
QFish Reservoirs	1	
HydropowerTotalDischargeCPs	7	
HydropowerOutflowCPs	7	
Kaw Comp Incrs	6	Computational
Keystone Comp Incrs	4	Computational

3 of 22 Subbasins selected (22 in model)

Order: Model

☑ Include Non-Computational User-Defined Subbasins

☑ Include Object Membership List

Clicking the “Add Subbasins” button brings up Subbasin Selector. The items selected with the selector which are not already in the Subbasins list are added to the list.

The selected Subbasins are removed from the list by clicking the “Remove Selected” button.

Supported orders, selected with the “Order” ComboBox are:

- Model
- Name

The **Model order** is the order of subbasins as they appear in the Subbasin manager.

The **Name order** is a the ascending alphabetic order by subbasin name.

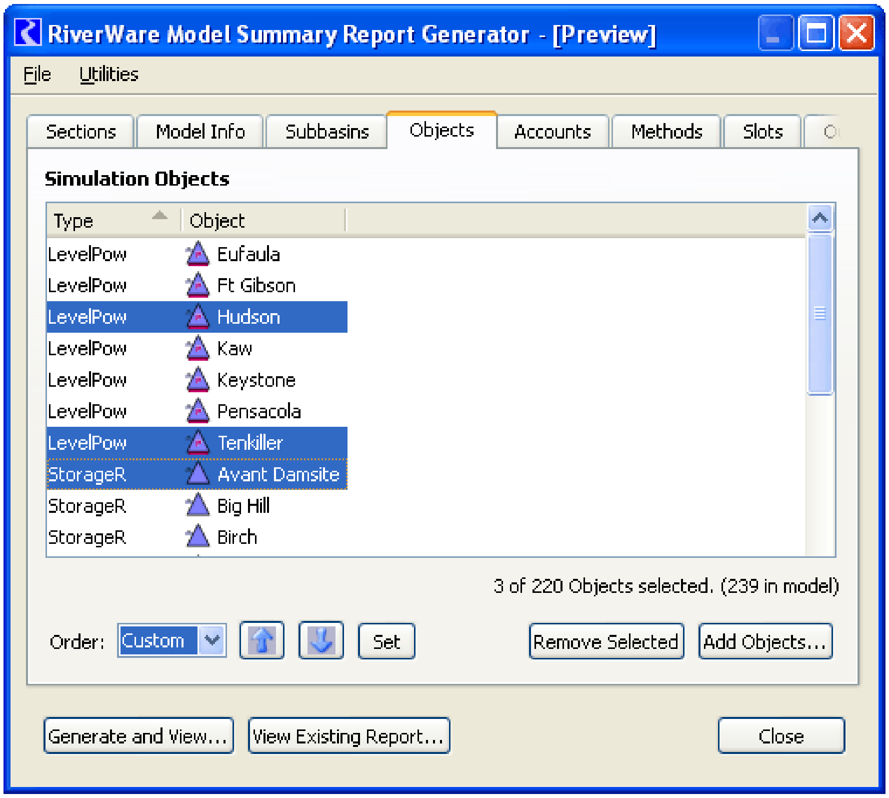
The **“[X] Include Non-Computational User-Defined Subbasins”** checkbox allows the user to have all user-defined subbasins to be included in the report, or limit included subbasins to Computational Subbasins. Note that Computation Subbasins are the only type of Subbasins which have methods and slots. Other types of subbasins consist only of an ordered list of Simulation Objects.

The **“[X] Include Object Membership List”** checkbox causes Subbasin instance sections in the report to include the subbasin’s list of Simulation Objects.

Subbasin Methods and Slots are included in Subbasin instance sections if those are checked (for Subbasin) on the Sections Tab *(described in a prior section).* The specific Subbasin methods and slots included in the report are configured respectively on the Methods and Slots Tabs, using the Subbasin Methods and Subbasin Slots Sub-Tabs. *(Read more in subsequent sections).*

2.6 Config Dialog Tab 4: Objects

The Objects Tab indicates which, and in what order Simulation Objects appear in the Objects section of the generated report document.



Type	Object
LevelPow	Eufaula
LevelPow	Ft Gibson
LevelPow	Hudson
LevelPow	Kaw
LevelPow	Keystone
LevelPow	Pensacola
LevelPow	Tenkiller
StorageR	Avant Damsite
StorageR	Big Hill
StorageR	Birch

3 of 220 Objects selected. (239 in model)

Order: Custom

All the controls are mostly analogous to the same controls on the Subbasins Tab.

Supported SimObj order options are:

- Custom
- Model
- Name
- Type / Name

The **Custom order** is an order created in this dialog, in the displayed list. Selected items can be moved up or down with the blue buttons. The user can click on any of the column headers to sort by the data in that column -- clicking again reverses the order. The displayed order (e.g. as a result of clicking in a column header) can be made the Custom order by clicking the “Set” button. The Set button is active only when the displayed order differs from the Custom Order. The blue arrow buttons are active only when the “Custom” order is selected and when movement of the selected items is possible in the respective direction (e.g. the UP arrow is disabled if the item selection includes the topmost item).

The **Model order** is the order of Simulation Objects as defined within the RiverWare mode.

The **Name** order is sorted by **Object Name.**

The **Type / Name order** is first sorted by the Object Type (polymorphically, with all Reservoirs grouped together), and then by Object Name.

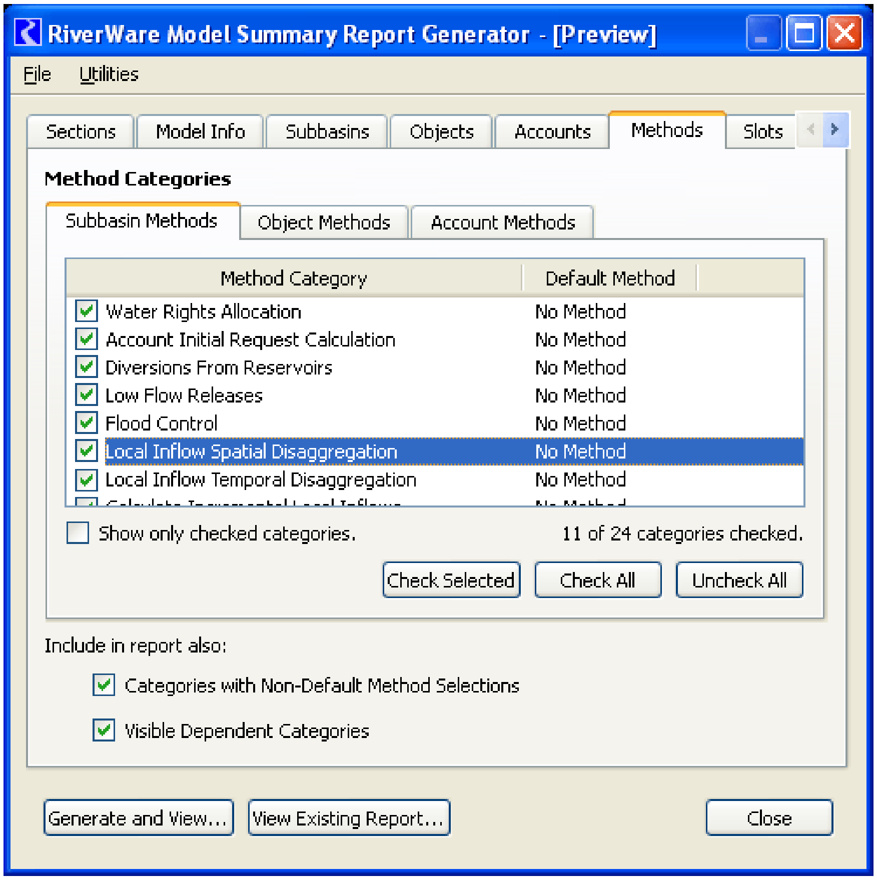
Simulation Object Methods and Slots are included in Object instance sections if those are checked (for Objects) on the Sections Tab *(described in a prior section)*. The specific Object methods and slots included in the report are configured respectively on the Methods and Slots Tabs, using the Object Methods and Object Slots Sub-Tabs. *(Read more in subsequent sections).*

2.7 Config Dialog Tab 5: Accounts

The Accounts Tab *(not illustrated with a mock-up image)* is similar to the Objects Tab. Individual Account instances are selected to appear in the report -- either within Object instance sections, or grouped together after the Object instance sections (depending on the setting on the Sections Tab; *described in a prior section).*

2.8 Config Dialog Tab 6: Methods

The Methods Tab defines which Method Categories show up in the generated report.



Method Categories

Method Category	Default Method
☑ Water Rights Allocation	No Method
☑ Account Initial Request Calculation	No Method
☑ Diversions From Reservoirs	No Method
☑ Low Flow Releases	No Method
☑ Flood Control	No Method
☑ Local Inflow Spatial Disaggregation	No Method
☑ Local Inflow Temporal Disaggregation	No Method

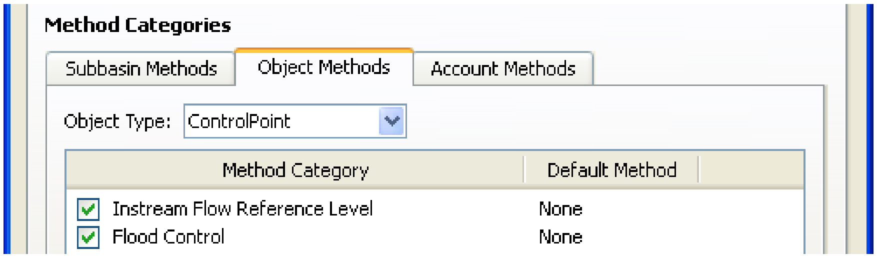
☐ Show only checked categories.

11 of 24 categories checked.

Include in report also:

☑ Categories with Non-Default Method Selections

☑ Visible Dependent Categories



Method Categories

Object Type: ControlPoint

Method Category	Default Method
☑ Instream Flow Reference Level	None
☑ Flood Control	None

A list of "category entities" is automatically built from the relevant objects (Subbasins, Simulation Objects, or Accounts) selected to be in the report (on prior tabs).

A "category entity" list item represents actually all of the same-named categories on all of the objects of a particular type of object.

A checked method entity indicates that method is included in the report with all objects which have that category.

If the Subbasin Methods Sub-Tab is selected , the "category entity" list is built from the Subbasins to appear in the report.

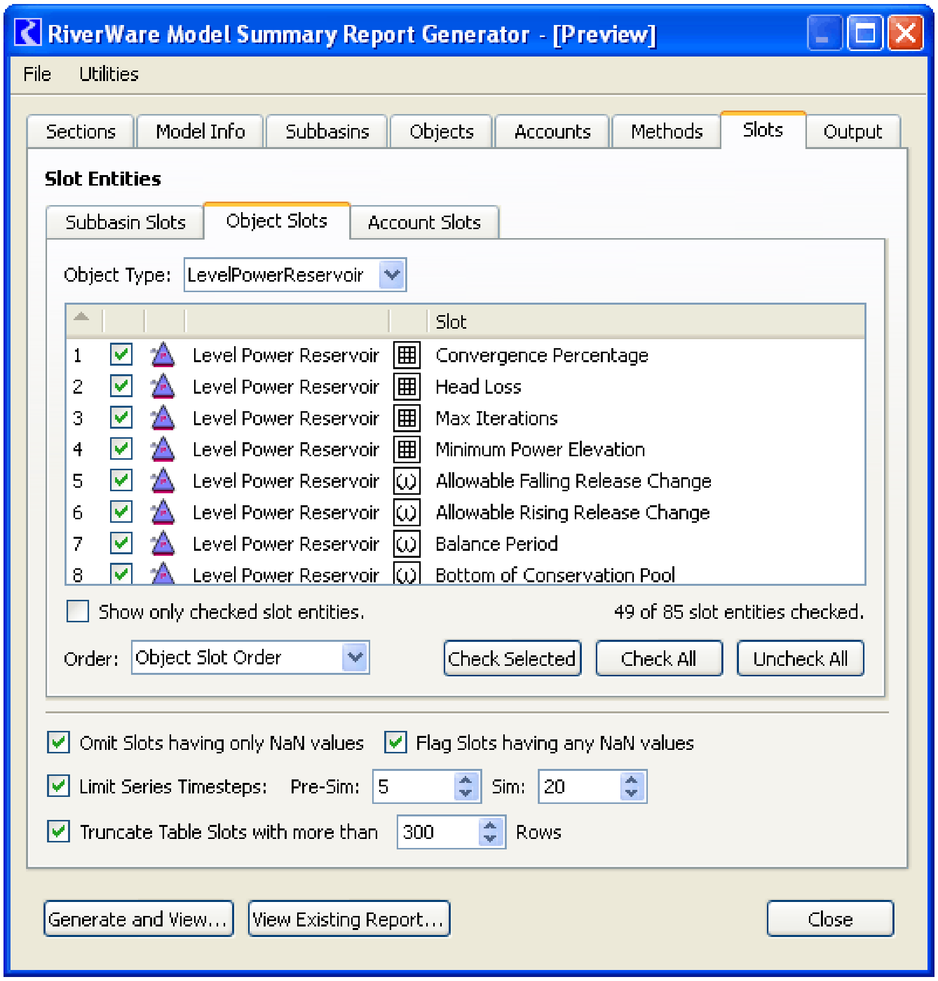
The Object Methods and Account Methods Tabs have an additional control: a combo box to select a specific Simulation Object or Account type. The list of methods is built from all the Objects or Accounts of the selected type which have been selected to appear in the report.

The two checkboxes below the list -- under "Include in report also..." -- provide an alternate way for additional Method Categories to be included in the report -- on a *per object* (Subbasin, SimObj, or Account) *basis.*

- If the **"[X] Categories with Non-Default Method Selections"** checkbox is checked, then -- for any specific object -- categories which have a non-default method selected are included (regardless of any settings in the "category entity list".
- Similarly, if the **"[X] Visible Dependent Categories"** checkbox is checked, those Categories are also included.

2.9 Config Dialog Tab 7: Slots

The Slots Tab defines many aspects of how Slots show up in the generated report:



Slot Entities

Object Type: LevelPowerReservoir

					Slot
1	☑	▲	Level Power Reservoir	▦	Convergence Percentage
2	☑	▲	Level Power Reservoir	▦	Head Loss
3	☑	▲	Level Power Reservoir	▦	Max Iterations
4	☑	▲	Level Power Reservoir	▦	Minimum Power Elevation
5	☑	▲	Level Power Reservoir	ω	Allowable Falling Release Change
6	☑	▲	Level Power Reservoir	ω	Allowable Rising Release Change
7	☑	▲	Level Power Reservoir	ω	Balance Period
8	☑	▲	Level Power Reservoir	ω	Bottom of Conservation Pool

□ Show only checked slot entities. 49 of 85 slot entities checked.

Order: Object Slot Order

☑ Omit Slots having only NaN values ☑ Flag Slots having any NaN values

☑ Limit Series Timesteps: Pre-Sim: 5 Sim: 20

☑ Truncate Table Slots with more than 300 Rows

1. Which Physical and Account Slot *entities* (names of Slots relative to object types) are included.
2. Whether or not selected Slots should be included even if they have no values (all "NaN").
3. Whether the existence of any NaN values should be "flagged" with special notices. Such notices appear within the Slot Instance detail section *and* in the Report Generation Status section.
4. For SeriesSlots selected to be in the report, how many Pre-simulation and Simulation timesteps should be included, if not all timesteps.
5. For Table Slots having many rows, whether such tables should be truncated if the row count exceeds a specified quantity. When a Table Slot is truncated, only its first and last several rows are shown.

The latter two settings are provided because Series Slots and Table Slots can be quite large, and, in the case of Series Slots, it will sometimes be important to show initialization data (generally, pre-simulation timesteps). It may also be useful to show only representative values of large Table Slots. [See Question 4 in the Draft USACE Text Output requirements document: *"Do we need to be able to show initial reservoir storage, PE or other initial conditions?"*].

Similar to the inclusion of Method Categories in a report *(see prior section),* Slots are identified to be included by “Slot Entity” rather than by “Slot Instance”. A “Slot Entity” represents similarly -named Slots on the set of objects of one particular type. The displayed list of “Slot Entities” is automatically generated from the set of objects (Subbasins, SimObjs, or Accounts) selected to be in the report (on prior tabs). Only those “Slot Entities” which are checked are included in the report.

Supported Slot Entity orders include:

- **Object Slot Order** -- the Slot order saved with the SimObj (or CompObj) instance.
- **Object Type Order** -- the Slot order saved with the SimObj type.
- **Slot Name** -- the ascending alphabetic order by Slot name.
- **Slot Type / Name** -- sorted first by Slot Type, and then by Slot name.
- **Unit Type / Name** -- sorted first by Unit Type, and then by Slot name.

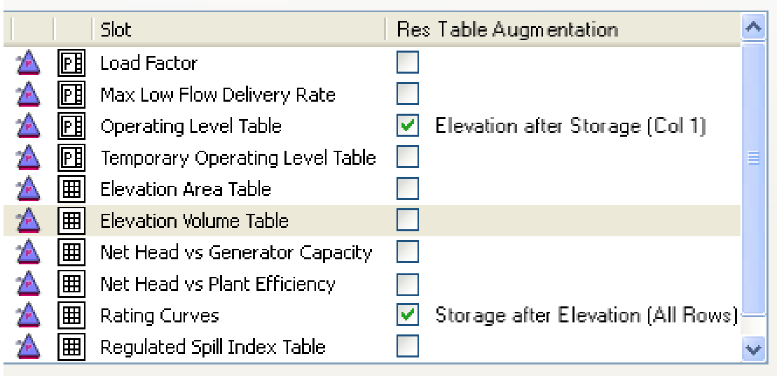
The text note under the Slot Name list will indicate information about the number of Slots and Slot Names in three parts: (1) the list selection, (2) selected for the report, and (3) in the model. For example:

> LevelPowerReservoir Objects: Sel: 7 slots (1 name); Report: 77 slots (11 names); Model: 77 slots (11 names).

Support for Reservoir Table Augmentation:

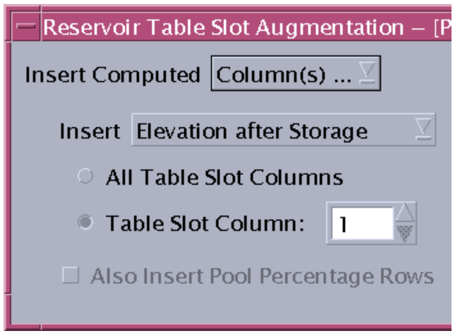
A special feature for Reservoirs is supported in the Slot Entity list. When a Reservoir Simulation Object Type is selected, the list includes an additional column: **“Res Table Augmentation”**.

Reservoir Table Augmentation can be applied to Reservoir Table Slots including any of the three entities related to each other through the Reservoir Elevation Volume Table Slot or the Elevation Area Table.



Slot	Res Table Augmentation
Load Factor	☐
Max Low Flow Delivery Rate	☐
Operating Level Table	☑ Elevation after Storage (Col 1)
Temporary Operating Level Table	☐
Elevation Area Table	☐
Elevation Volume Table	☐
Net Head vs Generator Capacity	☐
Net Head vs Plant Efficiency	☐
Rating Curves	☑ Storage after Elevation (All Rows)
Regulated Spill Index Table	☐

Clicking ON a toggle button brings up the Reservoir Table Augmentation dialog box. *See more details in a subsequent section.*



Reservoir Table Slot Augmentation – [P

Insert Computed: Column(s) ...

Insert: Elevation after Storage

○ All Table Slot Columns

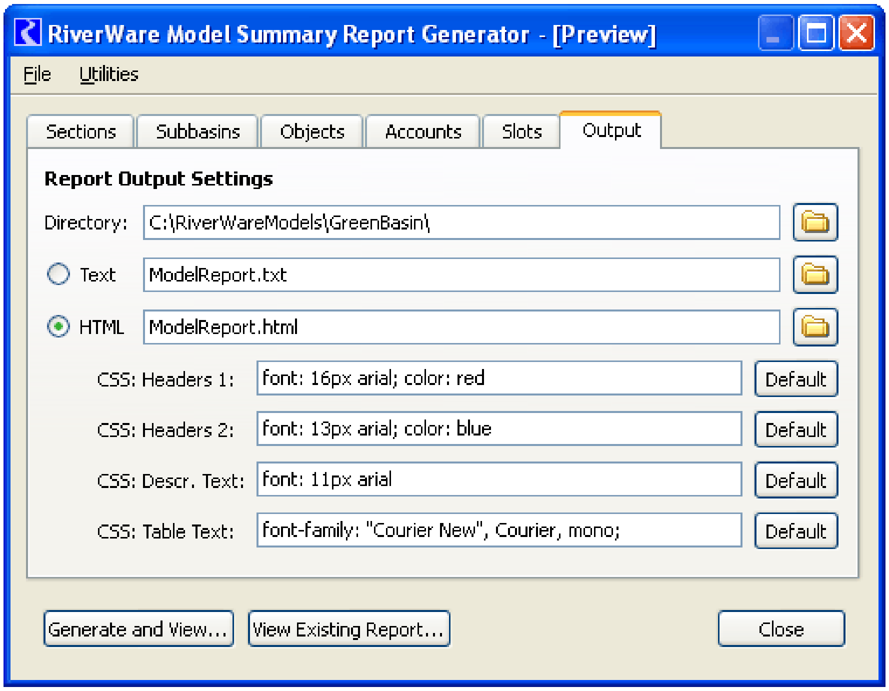
● Table Slot Column: 1

☐ Also Insert Pool Percentage Rows

2.10 Config Dialog Tab 8: Output

The Output Tab is used for:

1. Determining the file path of the generated report.
2. Determining the file type of the generated report: either plain text or HTML.
3. In the case of HTML, configuring text attributes for headers, descriptive text and tabular text. Text attributes are entered by the user using CSS (HTML cascading style sheet) syntax.



Report Output Settings

Directory: C:\RiverWareModels\GreenBasin\

○ Text ModelReport.txt

● HTML ModelReport.html

CSS: Headers 1: font: 16px arial; color: red [Default]

CSS: Headers 2: font: 13px arial; color: blue [Default]

CSS: Descr. Text: font: 11px arial [Default]

CSS: Table Text: font-family: "Courier New", Courier, mono; [Default]

Additionally, a little editor dialog box can be developed to fill in common CSS properties. *(Hooks for a CSS editor are not shown in the accompanying mock-up image).*

3.0 Reservoir Table Augmentation / Dialog

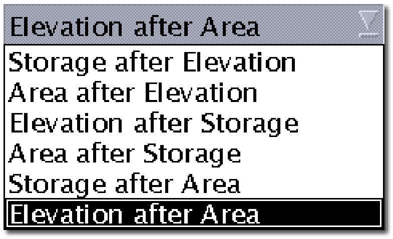
In general, for each Slot included in the report, output for only the Slot's attributes and contained data is included.

An exception to this is the addition of computed rows or columns within Reservoir Table Slots related to (1) elevation, (2) storage volume or (3) surface area.

Special computed rows or columns can be inserted after Table Slot rows or columns containing any of those three entities. These will contain data computed from the Table Slot rows or columns and data in -- or interpolated from -- the Reservoir's **Elevation Volume Table Slot** and **Elevation Area Table Slot.**

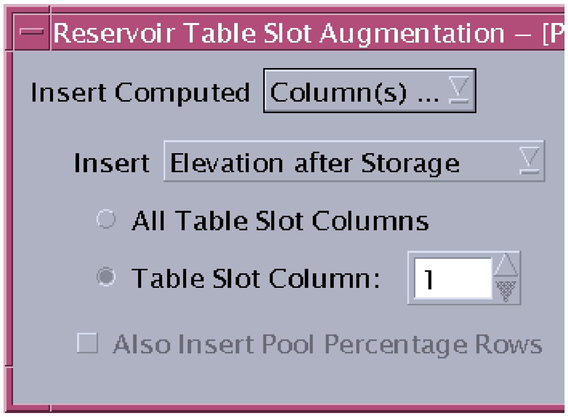
There are two types of support, described in the following sections.

1. Single Row or Column Insertion
2. All-Row or All-Column Insertion



3.1 Single Row or Column Insertion

For Reservoir Table Slots with a single row or column having the relevant data (elevation, storage, or surface area), **a single computed row or column** can be inserted after that Table Slot row or column.


Reservoir Table Slot Augmentation – [P

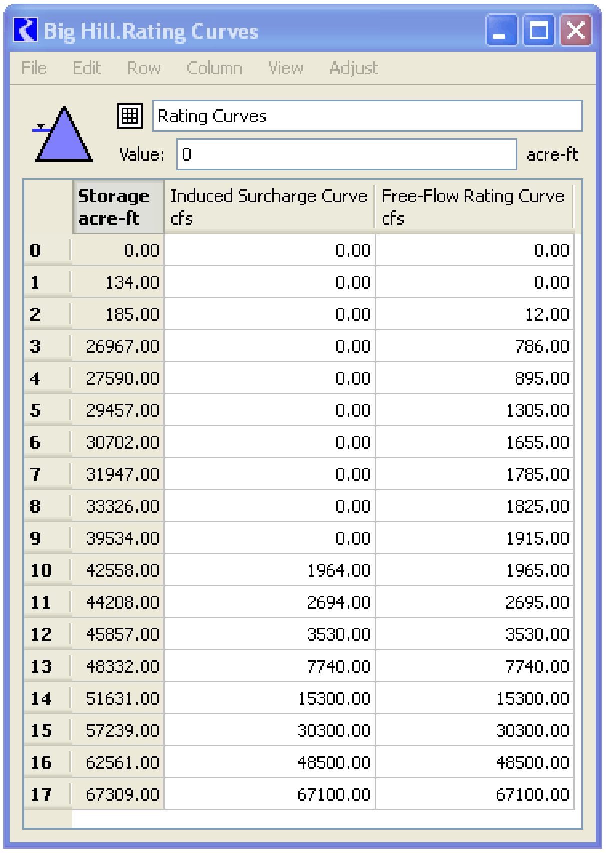
Insert Computed: Column(s) ...

Insert: Elevation after Storage

○ All Table Slot Columns

● Table Slot Column: 1

□ Also Insert Pool Percentage Rows



Rating Curves

Value: 0 acre-ft

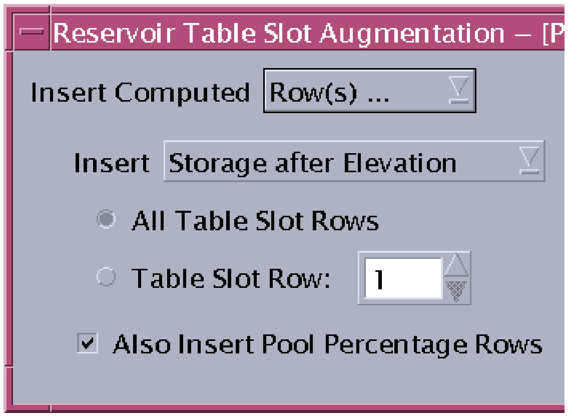
	Storage acre-ft	Induced Surcharge Curve cfs	Free-Flow Rating Curve cfs
0	0.00	0.00	0.00
1	134.00	0.00	0.00
2	185.00	0.00	12.00
3	26967.00	0.00	786.00
4	27590.00	0.00	895.00
5	29457.00	0.00	1305.00
6	30702.00	0.00	1655.00
7	31947.00	0.00	1785.00
8	33326.00	0.00	1825.00
9	39534.00	0.00	1915.00
10	42558.00	1964.00	1965.00
11	44208.00	2694.00	2695.00
12	45857.00	3530.00	3530.00
13	48332.00	7740.00	7740.00
14	51631.00	15300.00	15300.00
15	57239.00	30300.00	30300.00
16	62561.00	48500.00	48500.00
17	67309.00	67100.00	67100.00

This configuration would result in the insertion of a "Elevation" column after the first ("Storage") column. The Elevation values will be taken from -- or interpolated from -- values in the Reservoir's Elevation Volume Table Slot.

The referenced Table Slot Column (in this example, Column 1), only has to have Volume units. It does not have to be a slot column actually labeled "Storage". An error is reported if the referenced Table Slot Column does not have Volume units.

3.2 All-Row or All-Column Insertion

For Reservoir Table Slots having homogenous data of one of the three supported entities, a computed row or column can be inserted **after every Table Slot row or column.**



Reservoir Table Slot Augmentation – [P

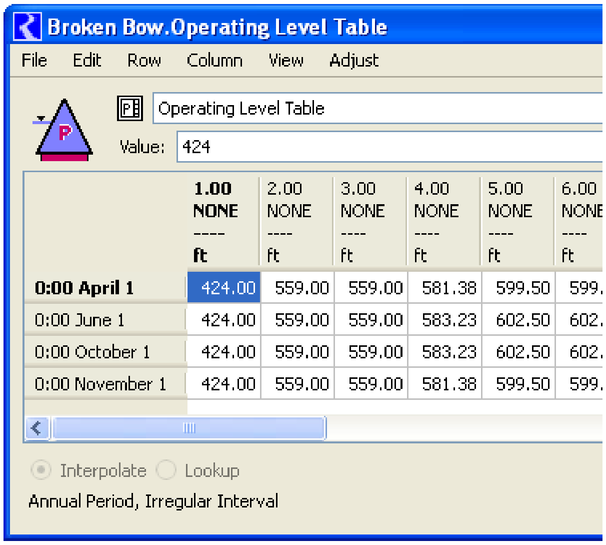
Insert Computed: Row(s) ...

Insert: Storage after Elevation

● All Table Slot Rows

○ Table Slot Row: 1

☑ Also Insert Pool Percentage Rows



Broken Bow.Operating Level Table

Operating Level Table

Value: 424

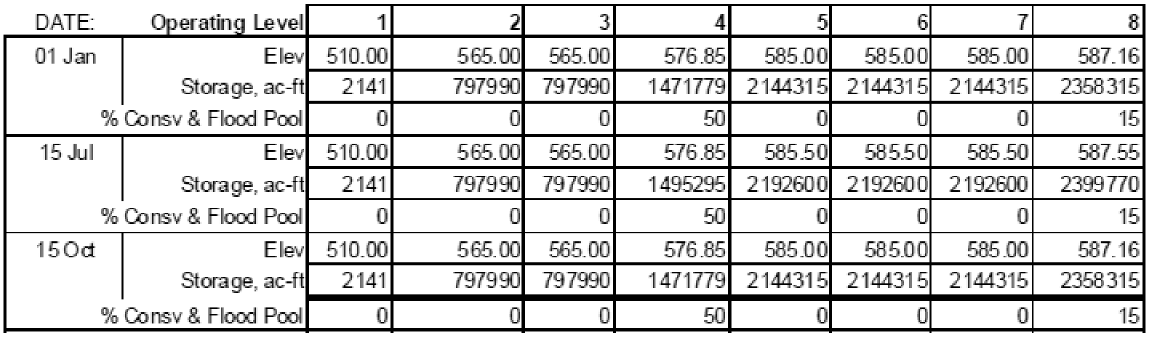
	1.00 NONE ----- ft	2.00 NONE ---- ft	3.00 NONE ---- ft	4.00 NONE ---- ft	5.00 NONE ---- ft	6.00 NONE ---- ft
0:00 April 1	424.00	559.00	559.00	581.38	599.50	599.
0:00 June 1	424.00	559.00	559.00	583.23	602.50	602.
0:00 October 1	424.00	559.00	559.00	583.23	602.50	602.
0:00 November 1	424.00	559.00	559.00	581.38	599.50	599.

● Interpolate ○ Lookup

Annual Period, Irregular Interval

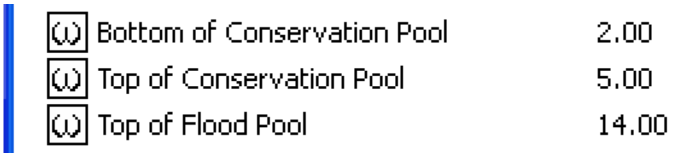
This configuration would result in extra rows being inserted after every Table Slot row. The values within the table represent elevations [ft]. The added rows would be Storage (volume) rows.

The "**[X] Also Insert Pool Percentage Rows"** checkbox generates one or two (probably two) super-extra rows for each Table Slot row to indicated the percentage of the represented Storage quantity with respect to the Conservation Pool and the Flood Pool. See the specification for this in the USACE Text Output requirements document. The resulting output is illustrated in that document with this mock-up (which shows only one row for both Conservation Pool and Flood Pool data -- but this will likely be split into two distinct rows):



DATE:	Operating Level	1	2	3	4	5	6	7	8
01 Jan	Elev	510.00	565.00	565.00	576.85	585.00	585.00	585.00	587.16
	Storage, ac-ft	2141	797990	797990	1471779	2144315	2144315	2144315	2358315
	% Consv & Flood Pool	0	0	0	50	0	0	0	15
15 Jul	Elev	510.00	565.00	565.00	576.85	585.50	585.50	585.50	587.55
	Storage, ac-ft	2141	797990	797990	1495295	2192600	2192600	2192600	2399770
	% Consv & Flood Pool	0	0	0	50	0	0	0	15
15 Oct	Elev	510.00	565.00	565.00	576.85	585.00	585.00	585.00	587.16
	Storage, ac-ft	2141	797990	797990	1471779	2144315	2144315	2144315	2358315
	% Consv & Flood Pool	0	0	0	50	0	0	0	15

The **"[X] Also Insert Pool Percentage Rows"** operation depends on these three scalar slots on the Reservoir Object having defined values . They represent Operating Levels (the columns of the table shown above).



Bottom of Conservation Pool	2.00
Top of Conservation Pool	5.00
Top of Flood Pool	14.00

4.0 Report Generation

Three report file formats are discussed in subsequent sections:

- Supported: Plain text (ASCII)
- Supported: HTML
- **NOT Supported:** XML (but see requirements for requirements).

Some aspects of plain text generation is actually more difficult to implement than HTML generation, because -- for text presented in tables, or otherwise aligned in columns -- the generator must explicitly calculate horizontal geometry and render row and column dividers (with text characters). Also, paragraphs must be explicitly wrapped.

The various report sections -- most of which *conditionally* appear in the report (as specified on the Sections Tab in the report configuration dialog) -- are described in the subsequent document sections.

- Report Header and Contents (first section of the report)
- Model File Info
- User Model Comments
- Subbasins
- Simulation Objects
- Accounts
- Slots (subsections within other sections)
- Report Generation Status

Generally, for numeric quantities accompanied by a wordy description, we propose that **the number come first, followed by the description.** This would make it reasonable to present lists of such entities in two columns, with the numeric quantity in a relatively narrow initial column. [NOTE: This differs from the convention used in the USACE SUPER report].

In general, in the report, **1x1 Table Slots** will be presented “as” **Scalar Slots:** when enumerated as table rows, their value would be shown. However, of course, their correct Slot type would be indicated, and, -- in the HTML report -- they would be distinguished using their respective Slot Type *icons* (if those are shown).

4.1 Plain Text (ASCII) Output

Plain text will be formatted with the assumption that the file is presented in a monospaced font (e.g. `Courier`). A font specification is *not* associated with the text file -- that is a property of the plain text viewer being used.

Tables -- and any presentation involving multiple rows of fixed field widths -- need to be processed in two passes: the maximum field width of each column needs to be measured (in number of characters) before the first row can be generated. Columns with numeric data will be right justified.

Lines of text, and paragraphs will be wrapped at 78 characters. There will be no attempt to *strictly* limit the character width of Tables. We may, however, choose to omit certain columns of non-essential or redundant data from the text report.

Only printable ASCII characters will be used, with the exception of Ctrl-L for page breaks. Tab characters will not be used, as their use does not have a specific definition. (Sometimes, a tabstop at every 8 characters is assumed, but that is fragile).

> *QUESTION:* Should the use of page break characters (Ctrl-L) be configurable. Minimally, they will be used after major sections, but they could optionally be inserted after selected types of subsections.

Table rows and columns will be delineated with these four characters: `'+' '-' '|' and '='` in a manner illustrated with this example:

```
+---+------------+-------------------------+------------------------+
|   | Storage    | Induced Surcharge Curve | Free-Flow Rating Curve |
|   | acre-ft    | cfs                     | cfs                    |
|===|============|=========================|========================|
| 0 |       0.00 |                    0.00 |                   0.00 |
| 1 |  427485.00 |                    0.00 |               14500.00 |
| 2 |  439168.00 |                  600.00 |               15200.00 |
| 3 |  451267.00 |                 1900.00 |               17800.00 |
| 4 |  482200.00 |                 6600.00 |               23050.00 |
| 5 |  568108.00 |                21400.00 |               39000.00 |
| 6 |  624480.00 |                47700.00 |               47700.00 |
| 7 |  778330.00 |                78800.00 |               78800.00 |
| 8 | 1149840.00 |               151500.00 |              151500.00 |
| 9 | 1187700.00 |               158770.00 |              158770.00 |
+---+------------+-------------------------+------------------------+
```

4.2 HTML Output

HTML output will support the user supplied **CSS (Cascading Style Sheet) styles** for the following uses, as specified on the Output Tab of the Model Summary Report Configuration Dialog:

- Header 1: Major Section Headers
- Header 2: Subsection Headers
- Descriptive Text: non-tabular text output
- Table Text: tabular text output (generally, should be a monospaced font).

HTML named anchors will be generated for all named RiverWare objects represented by a detail section within the report, and internal **links** to those named anchors will be implemented on *every occurrence* of the corresponding object (SimObj, Account, or Slot) name. For example, the list of Slots included at the head of a SimObj section will include links on the listed Slot names to the individual Slot Instance sections for each of those Slots.

Icon images are shown in the HTML report to indicate **Simulation Object types, Account types,** and **Slot types.** In HTML, images are references to distinct image files. The report generator function writes out those files (as PNG image files) to an automatically created **“reportImages”** subdirectory in the directory to which the report output file is written. When copying an HTML report file, the accompanying image subdirectory (folder) should also be copied. *[This has been implemented in the RiverWare Model Report Generation Demo]*.

4.3 XML Output -- NOT SUPPORTED.

An **XML schema** would need to be defined. Tentatively, this would include:

- Distinct XML elements for the supported RiverWare object types (Subbasins, Simulation Objects, Accounts, Slots),
- Elements or attributes for the types of those object types (SimObj types, Account types, Slot types)
- For Slots, an element for each column (with at least one column represented), with subelements or attributes for:
    - column label (if multiple columns exist on the slot), or:
    - numeric column value (optional; with scale and units)
    - display units (with scale)
    - display precision
    - minimum value, maximum value, convergence value (with scale / unit spec, if not in display units).
    - For Slots with Series, the start and end
- For Slots with named (text) row labels, elements for row labels.
- Elements for Slot data. Attributes for numeric table cells, if used, should be terse, for file size considerations.
- *... more analysis would be need to complete this list.*

If an XML schema would be supported, we should provide an **XSLT (or CSS) style sheet** to generate an HTML presentation from the generated XML report file, somewhat similar to the direct HTML generated report (though not necessarily as fully developed, e.g. vis-a-vis internal HTML links to named anchors, and probably lacking much of the redundancy -- see note below). Users would be welcome to adapt the XSLT style sheet for their own purposes.

Also, a development task of defining a *formal schema* (DTD, or something newer) should be included.

A note about redundancy: A canonical XML schema does not contain redundancy -- similar to the idea of "normalization" in database schema design. XML tools (like XSLT) have the ability to compile summary statistics from the detail data. However, for the purpose of translating the XML output to a presentation format (typically, HTML), it would be much more convenient to included redundant summary data in the generated XML. We could allow all but the most arcane summary data to be represented redundantly, but such elements should generally be defined as *optional* in the formal schema. Doing so would be helpful if the report output schema is ever adapted as a RiverWare Model Data exchange format.

We should be very careful to not over-design an XML schema for the purpose of report output. Even if we are interested in developing a schema also for RiverWare Model Data exchange, tools could be created to create a model data exchange XML document from a report output XML document. That is, distinct schema, optimized for the two data applications, could be used.

XML output is not explored further in this document -- with the exception of the possible use of XML to represent the RiverWare Model Report Configuration serialization (as an alternative to Flex/Bison).

4.4 Report Section: Header and Contents

- Model Name and Model File Path
- Date and time of report generation.
- List of the major sections within the report. The HTML version implements links to the various sections.

4.5 Report Section: Model File Info

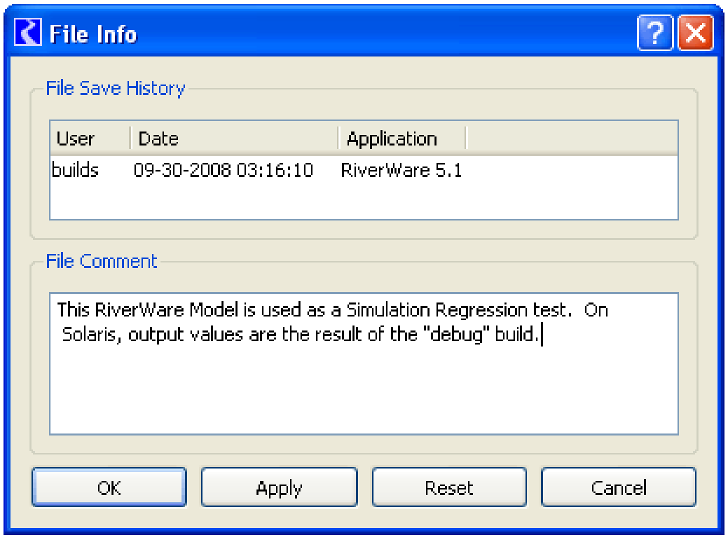
This section will include:

- Table: Simulation Object (SimObj) Types represented within the model -- one Object Type per row. Columns:
  - Object Type (name)
  - Number of SimObjs of that Object Type in the model.
  - Number of SimObjs of that Object Type in the report.
  - Number of Accounts under SimObjs of that Object Type in the report.
- Table: Account Types represented in model (if any). Columns:
  - Account Type (name)
  - Number of Accounts of that Account Type in the model.
  - Number of Accounts of that Account Type in the report.
- Controller Selection
- Run Period

4.6 Report Section: User Model Comments

This section includes the two types of information shown in the RiverWare File Info dialog box *(see image)*.

- Model File User Comments (paragraphs)
- Table: File Save History. Columns:
  - User
  - Save Date and Time,
  - RiverWare Version.



User	Date	Application
builds	09-30-2008 03:16:10	RiverWare 5.1

This RiverWare Model is used as a Simulation Regression test. On Solaris, output values are the result of the "debug" build.

4.7 Subbasins

4.7.1 Subbasins Summary Section

- Number of Subbasins in Model
- Number of Subbasins in Report
- Table with one row per Subbasin in the report. Columns:
  - Subbasin Name (HTML version implements link to the specific Subbasin section)
  - Subbasin Type (“Computational”)
  - Number of Simulation Objects
  - Number of Slots in Model (in the Subbasin)
  - Number of Slots in the report (in the Subbasin)

4.7.2 Subbasin Instance Sections

One instance *for each Subbasin,* including:

- **Subbasin Instance Summary Info** (repeat of the information in the *corresponding row* in the summary table from above).
- Table with list of **Simulation Objects within the Subbasin,** Columns:
  - Simulation Object Type (HTML implements SimObj Type *Icon* and *terse* Type Name)
  - Simulation Object Name (HTML implements link to the Simulation Object Instance section)
- Table with **Subbasin Methods (for Computational Subbasins),** conditionally excluding Method Categories with default Method selections. Columns:
  - Method Category
  - Selected Method for that Category
  - **QUESTION:** Do the Slots related to the Methods also need to be listed? If so, do so here in additional rows, indented (sort of).
- Table with Subbasin Slots, one Slot per row. Columns *(possibly abbreviated in Text output format):*
  - Slot Type (HTML implements SimObj Type *Icon* and terse Type Name)
  - Slot Name (HTML implements a link to the Slot Instance section)
  - For Scalar Slots (and 1x1 Table Slots): Slot Value
  - For Scalar Slots (and 1x1 Table Slots): Display Unit and Value
  - ... or Single-Column Slots: Unit Type
  - For Table Slots: Row and Column Count
  - For SeriesSlots: Start Date/Time and Number of Timesteps
  - Number of Non-Nan Values
- Conditionally, a **Slots Instance section** (defined below), if the Subbasin’s Slots are to be included within the Subbasin Instance Section.

4.8 Simulation Objects

The Simulation Objects section has an organization very similar to the Subbasins Section (but lacking the composition of contained Simulation Objects), and including an Accounts section, if applicable.

4.8.1 Simulation Objects Summary Section

- Number of SimObjs in Model
- Number of SimObjs in Report
- Table with one row per SimObj in the report. Columns:
    - •SimObj Type (HTML implements SimObj Type *Icon* and *terse* Type Name)
    - •SimObj Name (HTML version implements link to the specific SimObj section)
    - •Number of Accounts in Model (in the SimObj)
    - •Number of Accounts in the report (in the SimObj)
    - •Number of Slots in Model (in the SimObj)
    - •Number of Slots in the report (in the SimObj)

4.8.2 SimObj Instance Sections

One instance section *for each SimObj*, containing ...

- **SimObj Instance Summary Info** (repeat of the information in the *corresponding row* in the summary table, see above).
- If applicable, Table with list of **Accounts within the SimObj,** Columns:
    - •Account Type (HTML implements Account Type *Icon* and *terse* Type Name)
    - •Account Name (HTML implements link to the Account Instance section)
    - •Number of Supplies TO and FROM the Account
- Conditionally, an **Accounts Instance section** (defined below), if the SimObj's Accounts are to be included within the SimObj Instance Section.
- Table with **SimObj Methods,** conditionally excluding Method Categories with default Method selections. Columns:
    - •Method Category
    - •Selected Method for that Category
    - •**QUESTION:** Do the Slots related to the Methods also need to be listed? If so, do so here in additional rows.
- Table with SimObj Slots, one Slot per row. Columns *(possibly abbreviated in Text output format):*
    - •Slot Type (HTML implements SimObj Type *Icon* and terse Type Name)
    - •Slot Name (HTML implements a link to the Slot Instance section)
    - •For Scalar Slots (and 1x1 Table Slots): Slot Value
    - •For Scalar Slots (and 1x1 Table Slots): Display Unit and Value ... or Single-Column Slots: Unit Type
    - •For Table Slots: Row and Column Count
    - •For SeriesSlots: Start Date/Time and Number of Timesteps
    - •Number of Non-Nan Values
- Conditionally, a **Slots Instance section** (defined below), if the SimObj's Slots are to be included within the SimObj Instance Section.

4.9 Accounts

The Accounts section has an organization similar to the SimObj section (but lacking a composition of contained Accounts).

4.9.1 Accounts Summary Section

- Number of Accounts in Model (or under the current SimObj)
- Number of Accounts in Report (if relevant, only those under the current SimObj).
- Table with one row per Account in the report. Columns:
  - Account Type (HTML implements Account Type *Icon* and *terse* Type Name)
  - Account Name (HTML version implements link to the specific Account section)
  - Number of Supplies TO this Account
  - Number of Supplies FROM this Account
  - Number of Slots in Model (in the Account)
  - Number of Slots in the report (in the Account)

4.9.2 Account Instance Sections

One instance section for each Account, containing the following fields:

- **Account Instance Summary Info** (repeat of the information in the *corresponding row* in the summary table, see above).
- Table of Supplies TO this Account, with an initial terse Supply Type column.
- Table of Supplies FROM this Account, with an initial terse Supply Type column.
- Table with **Account Methods,** conditionally excluding Method Categories with default Method selections. Columns:
  - Method Category
  - Selected Method for that Category
  - **QUESTION:** Do the Slots related to the Methods also need to be listed? If so, do so here in additional rows. *(Not sure: Do Account Methods have explicitly associated Slots?).*
- Table with Account Slots, one Slot per row. Columns *(possibly abbreviated in Text output format):*
  - Slot Type (HTML implements SimObj Type *Icon* and terse Type Name)
  - Slot Name (HTML implements a link to the Slot Instance section)
  - For Scalar Slots (and 1x1 Table Slots): Slot Value
  - For Scalar Slots (and 1x1 Table Slots): Display Unit and Value ... or Single-Column Slots: Unit Type
  - For Table Slots: Row and Column Count
  - For SeriesSlots: Start Date/Time and Number of Timesteps
  - Number of Non-Nan Values
- Conditionally, a **Slots Instance section** (defined below), if the Account's Slots are to be included within the Account Instance Section.

4.10 Slots

Slots configured to appear in the report (depending on checkbox settings on the Sections Tab) do so in two ways:

- As a line-item in a Slot List within the containing object instance section, and
- with a Slot Instance Section.

4.10.1 Slots Lists

Instance sections of objects which contain objects include a list of Slots, with the following columns -- *possibly abbreviated for text output.*

- SimObj Type (HTML implements SimObj Type *Icon* and *terse* Type Name)
- SimObj Name (HTML implements a link to the SimObj Instance section)
- Account Type (HTML implements Account Type *Icon* and *terse* Type Name)
- Account Name (HTML implements a link to the Account Instance section)
- Slot Type (HTML implements Slot Type *Icon* and *terse* Type Name)
- Slot Name (HTML implements a link to the Slot Instance section)
- For Scalar Slots (and 1x1 Table Slots): Slot Value
- For Scalar Slots (and 1x1 Table Slots): Display Unit and Value ... or Single-Column Slots: Unit Type
- For non-scalar Slots: Row and Column Count
- For SeriesSlots: Start Date/Time and Number of Timesteps
- Number of Non-Nan Values

4.10.2 Slot Instance Sections

Each Slot Instance section shows the following information, when applicable:

- Containing SimObj Type and Name (HTML implements type icon and link to SimObj instance)
- Containing Account Type and Name (HTML implements type icon and link to Account instance)
- For Scalar and 1x1 Tables: Slot Value
- For each Column (often, just one): Scale and Unit
- For each Column (often, just one): Minimum value, Maximum value, Convergence value / Scale & Units

Slot data is presented in a format appropriate for the particular Slot type, generally tabular.

- List Slots are shown as a list.
- Table Slots are shown as ... tables, possibly truncated (indicated clearly). Tables on Reservoirs may implement Reservoir Table Slot Augmentation. (Described above in this document).
- SeriesSlots show columns for timestep date/times, numeric values and Flag characters (where relevant). For multiple-column SeriesSlots, of course, the timestep date/time is shown only in the first column. Long SeriesSlots may be truncated (indicated clearly).

QUESTION: Should the HTML presentation show cell colors for Flags? Should this be configurable? Currently cell colors are defined only for Accounting Slots, and for Slots in the context of an SCT. (So, this would need to be worked out).

4.11 Report Generation Status

The report ends with a section which summarizes special conditions encountered during the generation of the report, for example:

1. a list of the Slots which were configured to be in the report, but which were omitted because they didn’t have any non-NaN values (if enabled for the report).
2. a list of Slots having any NaN values (if enabled for the report).
3. a list of the TableSlots and SeriesSlots which were truncated (according to configured size limits, and only if enabled for the report).

During development, and optionally for debug, certain unspecified diagnostic output will also be shown in this section.

After starting a report generation operation, some of this information will also be presented in a popup dialog informing the user that the report has been generated.

5.0 Report Configuration Persistence / Serialization

Two “serialization” technologies are being considered for report configuration objects -- that is, for representing report configurations within RiverWare Model Files and in stand-alone RiverWare Report Configuration Files.

- Flex/Bison, or
- XML (using either DOM or SAX import processing).

Flex/Bison is a technology we have a lot of experience with. This grammar-driven tool is used only for reading the serialization -- writing is done in plain C++ code. We currently deploy Flex/Bison-parsable objects within Tcl-based RiverWare Model Files and as stand-alone files (e.g. RplSets and SCT Configurations). This would be the quickest development option.

XML is a new technology for us, but it is something we will want to start using, possibly for an open specification for RiverWare Model data. The level of complexity of the RiverWare Report Generation Configuration makes it a good candidate for our first use of XML, but a little more time would be needed, being that this is our first exposure to the Qt XML tools. There are two general approaches, and Qt APIs, for reading XML files:

- **DOM (Document Object Model)** -- this represents a full in-memory representation of the XML document. This would be applicable for both saving and loading Report Generation Configuration objects. For saving, RiverWare would create a DOM object -- using the Qt DOM API -- and then cause it to be written out. For loading, a full DOM object would be created by loading it (from a long string handed over to us from Tcl, or from a file), and would be “inspected” internally within RiverWare.
- **SAX (Simple API for XML)** -- a lexical, event-driven technology, analogous to Tcl or Flex/Bison (but with better type safety). This addresses only loading (parsing) the Report Configuration object.

  Note: Writing the XML document would need to be addressed in some other way (either with plain C++ code, or using the DOM API). Even if SAX is used for loading (parsing), we should probably still use DOM for writing out the XML serialization. This would guarantee the validity of the XML file -- which is a *very big deal* with XML.

Advantages of DOM (for XML) over SAX are:

- The same technology is used for saving and loading (parsing) the serialization. This provides better encapsulation of the output format.
- DOM directly supports “random access” to the data, since the entire XML document is represented in memory at a single time.

The advantages of SAX (for XML) over DOM are:

- SAX, being a serial processor, doesn’t require the entire XML document to reside in memory at any one time. This may be important when using XML for an entire RiverWare model, but would not be a concern for the RiverWare Model Report Configuration. We would favor SAX probably only as an opportunity to prepare for XML representations of RiverWare Model files.

Note that this discussion of the use of XML to represent the Report Configuration is *unrelated* to the idea of generating XML for the actual *report document.* The latter is not being proposed in this design, but the *possibility* is discussed in a prior section: Report Generation: XML Output -- NOT SUPPORTED.

6.0 Enumeration of Fields within the RiverWare Model Report Configuration

This enumeration is presented for the following purposes:

1. To inform the decision of whether or not Report Configuration objects can reasonably be stored within RiverWare Models.
2. To help characterize the complexity of the Report Configuration object structure in order to support the time estimate given for development of the serialization.

This enumeration is suggestive of a specific serialization design. The actual implemented design may differ in probably-minor ways.

Fields are enumerated in the following parts:

1. General Fields (one instance)
2. Section Fields (one per Section)
3. SimObj Class Fields (one per SimObj Type, including CompObj)
4. Account Class Fields (one per Account Type)
5. Reservoir Slot Entity Fields (For each Reservoir Non-Scalar Slot Entity in report)

6.1 General Fields (one instance)

Value Type	Description
enum	Account Grouping: (1) Under Objects, (2) Separate Sections
bool	Include SimObj Type Statistics
bool	Include Account Type Statistics
bool	Include Controller Selection and Run Period
bool	Include Model File User Comments
bool	Include File Save History
int	File Save History entry count limit
string list	Subbasin instance name list (one item for each included user-defined subbasin)
enum	Subbasin order type (Model, Name)
bool	Include Non-Computational User-Defined Subbasins
bool	Include SimObj Membership List
string list	SimObj instance name list (one item for each SimObj) (ordered: representing the Custom order)
enum	SimObj order type (Custom, Model, Name, TypeName)
bool	Include also all Method Categories with Non-Default Method Selections
bool	Include also all Visible Dependent Categories
bool	Omit Slots having only NaN values
bool	Flag Slots having any NaN values
bool	Limit Series Timesteps ...
int	... Pre-Simulation Timestep Limit (e.g. 5)
int	... Simulation Timestep Limit (e.g. 20)

bool	Truncate Table Slots with many rows
int	... Table Slot truncation row limit (e.g. 300)
string	Report Output File Directory
string	Text Report Output File Name
string	HTML Report Output File Name
string	CSS for Header 1
string	CSS for Header 2
string	CSS Descriptive Text
string	CSS Table Cell Text

Totals in this section:

- enums: 3
- ints: 4
- bools: 13
- strings: (7) + (number of SubBasins) + (number of SimObjs)

6.2 Section Fields (one per Section for Five Sections)

Value Type	**Description**
string	Section Name (ModelInfo, ModelComments, Subbasins, Objects, Accounts)
int	Unique Ordinal (1...)
bool	Section Included in Report
bool	Include Methods (if applicable)
bool	Include Slots (if applicable)
enum	Object Slot Order (ObjSlotOrder, ObjTypeSlotOrder, SlotName, SlotTypeName, UnitTypeName)

Totals in this section:

- enums: 1
- ints: 1
- bools: 3
- strings: 1

6.3 SimObj Class Fields (one per SimObj Type, including CompObj)

Value Type	Description
string list	Included Method Category Names (currently, order is not significant)
string list	Included Slot Entities

Totals in this section:

- strings: (number of reported categories in SimObj Type) + (number of reported Slot Entities in SimObj Type).

6.4 Account Class Fields (one per Account Type: Storage, Diversion, InstreamFlow, PassThrough)

Value Type	Description
string list	Included Method Category Names (currently, order is not significant)
string list	Included Slot Entities

Totals in this section:

- strings: (number of reported categories in Account Type) + (number of reported Slot Entities in Account Type).

6.5 Reservoir Slot Entity Fields (For each Reservoir Non-Scalar Slot Entity in report)

These are arranged within sections which provide a SimObj or Account Type context.

Value Type	Description
string	Reservoir Slot Entity Name
bool	Res Table Augmentation Enabled
enum	Res Table Augmentation Mode (Undefined, All Slot Columns, One Slot Column)
int	Res Table Augmentation, Table Slot Column Index (for “One-Column” mode)

Totals in this section:

- bools: 1
- enums: 1
- ints: 1
- strings: 1

7.0 Serialization Size Estimate

This estimate is provided for the purpose of informing the decision of whether or not Report Configuration objects can reasonably be stored within RiverWare Models.

The *estimate* is based on:

- Enumeration of Fields within the RiverWare Model Report Configuration (prior section)
- The size of a report for a large model -- a large “arkansas” model -- assuming a maximum configuration.

Note: In the subsequent sections, all sizes are in units of bytes. Single-byte character encoding (generally, ASCII) is assumed.

7.1 Sample Model Counts

The following table provides counts of entities within the **ARKCOE9_SEP08.mdl** RiverWare model file. In order to compute an upper limit, it is assumed that **ALL User-Defined Subbasins, Simulation Objects and Slots** are included in the report.

Count	Description
93,736,515	[bytes] -- Zipped (compressed) model file size (89 MB)
230,506,932	[bytes] -- Unzipped (non-compressed) model file size (220 MB)
20	User-defined Subbasins
291	Simulation Objects (including CompObjs for Computational Subbasins)
24	Reservoirs
73	Reservoir Non-Scalar Slot Entities

SimObj Type Ordinal	Obj Count	Category Count	Slot Entity Count	Simulation Object Type
1	12	25	14	CompObj (for Computational Subbasins)
2	15	2	3	Confluence
3	53	19	37	Control Point
4	72	(N/A) 0	49	Data Object
5	14	3	7	Diversion Object
6	68	14	13	Reach
7	7	28	85	Level Power Reservoir
8	17	21	57	Storage Reservoir
9	33	10	13	Water User
(count: 9)	**291**	**122**	**278**	**TOTAL**

The quantities above are used as factors in computing the number of individual fields of various basic data types (bool, int, enumeration and string). (See table on the next page).

7.2 Serialization Size Estimate Calculations

Calculation Part / Data Type:			**bool count**	**int count**	**enum count**	**string count**
General Fields (x 1)			13	4	3	318
Section Fields (x 5)			15	5	5	5
	Category Count	**Slot Entity Count**				
Class Fields: CompObj	25	14	0	0	0	39
Class Fields: Confluence	2	3	0	0	0	5
Class Fields: Control Point	19	37	0	0	0	56
Class Fields: Data Object	0	49	0	0	0	49
Class Fields: Diversion Object	3	7	0	0	0	10
Class Fields: Reach	14	13	0	0	0	27
Class Fields: Level Power Res	28	85	0	0	0	113
Class Fields: Storage Reservoir	21	57	0	0	0	78
Class Fields: Water User	10	13	0	0	0	23
Account Class Fields (x 0)			0	0	0	0
Res Non-Scalar Slot Entity Fields (x 73)			73	73	73	73
Report Totals:			**101**	**82**	**81**	**796**

As stated above, in order to compute an upper limit, it is assumed that **ALL User-Defined Subbasins, Simulation Objects and Slots** are included in the report. The table above shows the numbers of resulting values of the basic data types (bool, int, enumeration and string) needed in the Report Configuration serialization.

In our **Flex/Bison encoding** many values show up as fields within multiple-field records (lines). But to reasonably insure an upper limit, we will assume that every value stands-alone in its own record, with a unique keyword. Similarly, for an **XML encoding,** we will assume that a full element (and not just an attribute) is used for every value -- this includes a start tag and an end tag, three “new-lines”, and some indentation.

Data Type	**Flex/Bison Average Data Size**	**Flex/Bison Record Keyword**	**Flex/Bison Record Size**	**Record Count (from above)**	**Size in Flex/Bison Serialization**
bool	1	15	16	101	1616
int	10	15	25	82	2050
enum	20	15	35	81	2835
string	20	15	35	796	27860
TOTAL:					**34361**

Data Type	XML Average Data Size	Average Start and End Tags	Other element overhead	Full Element Size	Record Count (from above)	XML Serialization
bool	5	35	10	50	101	5050
int	10	35	10	55	82	4510
enum	20	35	10	65	81	5265
string	20	35	10	65	796	51740
TOTAL:						**66565**

The table below shows the computations of percentages of the Report Configuration object sizes relative to specific model file examples. Uncompressed model sizes are used because the character counts for the Report Configuration serializations also represent uncompressed character (byte) counts.

Model	Uncompressed Model Size	Time Steps	Flex/Bison Serialization	F/Bison Size %	XML Serialization	XML Size %
ARKCOE9_SEP08.mdl *(source of counts used above)*	230,506,932	22281	34361	**0.015 %**	66565	**0.029 %**
CADSWES “arkansas” Regression Test	3,703,321	145	34361	**0.928 %**	66565	**1.797 %**

Note that actual Report Configuration size percentages are expected to be quite a bit lower, because *most* values -- in both Flex/Bison and XML encodings -- will be fields within multiple-field records (in XML, as attributes within elements rather than as their own elements). Also the actual average string and enumeration value size will likely be something like 10 or 12, not 20.

It’s reasonable to conclude that the *size* of either Report Configuration serialization (Flex/Bison, or XML) would not significantly impact the overall size of RiverWare models, if Report Configurations are included within models (i.e. as “Output Devices”). However, managing Report Configurations inside model files does require additional development cost, and that could be a reason for deciding against that design option.

8.0 Development Estimate

Much of the initial exploratory demo development work will be usable for a production quality report feature -- in particular, low-level tabular formatting in both Plain Text and HTML. [See the RiverWare Model Report Generation Demo document, 14 pages]. This addressed only formatting and output capabilities -- for a single Simulation Object -- including the creation of icon image files used in the HTML report and launching an external viewer application after the report document is created. None of this demo development addressed report configuration features.

TABLE 1. Development Tasks and Estimates

Task	Develop. Days	Optional / Alternative Days	Description
1			**Report Configuration Record Development**
1.1	1.0		Basic C++ class design and implementation
1.2	2.0		**Option 1:** Serialization Design and Implementation using **Flex/Bison**
1.3		2.0	**Option 2:** Serialization Design using XML
1.4		3.0	**Option 2:** Serialization Implementation: DOM output
1.5		3.0	**Option 2:** Serialization Implementation: DOM or SAX input
2			**Report Configuration Graphical User Interface (GUI)** (This is basically a “Report Configuration Record” editor).
2.1	1.5		Widget Layout with Qt Designer, starting from the work that has already been done to create the mock-up images for this document.
2.2	0.5		Section Tab, including reordering function.
2.3	0.5		Model Info Tab
2.4	1.0		Subbasins Tab, including adaptation of the Subbasin Panel from the Subbasin Editor for use in this new GUI.
2.5	1.5		Simulation Objects Tab
2.6	2.0		Account Tab. (Needs design completion).
2.7	2.0		Methods Tab, including compilation of a list of “Category Entities” from a set of Subbasins, SimObjs or Accounts.
2.8	2.0		Slots Tab, including compilation of a list of “Slot Entities” from a set of Subbasins, SimObjs or Accounts.
2.9	0.5		Output Tab -- with some filename assembly methods added to the Report Config class. Assuming just direct user entry of CSS for text attributes.
3			**Reservoir Table Augmentation**
3.1	1.5		Configuration dialog development.
3.2	1.5		Single-Row or Column Insertion “Joining” (Special Report Generation)
3.3	2.0		All-Row / All-Column Insertion “Joining” (Special Report Generation)

TABLE 1. Development Tasks and Estimates

4			**Report Generation (HTML and TEXT) -- see “Demo” note above.**
4.1	1.0		Model File Info & User Model Comments calculations and output
4.2	1.5		Subbasin Summary and Instance Sections
4.3	2.0		Simulation Object Summary and Instance Sections
4.4	2.0		Account Summary and Instance Sections
4.5	1.5		Slot Summary and Instance Sections COMPLETION (incl. Series Slots)
4.6	0.5		Report Generation Status Section
5			**Management of Report Generation Config Records as “Output Devices”**
5.1		3.0	Design and Implementation
6			**Completion**
6.1	1.0		Strategic use of CSS to minimize generated HTML file size. (No CSS was deployed in the Demo), including testing with multiple browsers. (Perhaps with a config switch, if any of the CSS seems edgy).
6.2	1.0		Usability testing and enhancements
6.3	1.0		Completeness and Correctness testing and fixes
6.4	2.0		Unknown Problem (code name: Flinch)
6.5	2.0		User Documentation
	35.0		**Total Development Estimate [Days] -- Not including options / alternatives**

--- (end) ---