RIVERWARE

Technical Documentation Version 7.3

Diagnostics

CU Center for Advanced Decision Support for
Water and Environmental Systems (CADSWES)
UNIVERSITY OF COLORADO **BOULDER**

Diagnostics
Table of Contents

1. Overview

Diagnostics are a powerful and helpful feature which provide you with the ability to gather useful messages during the course of a run or while building a model. User-selected diagnostic messages are displayed in the **Diagnostics Output Window** where Error and Warning messages also appear. Diagnostics may be filtered according to their context or turned off altogether. Errors and Warnings may not. Unlike Errors or Warnings, the **Diagnostics Output Window** will not be forced to the front by diagnostics. It is up to you to open the window.

1.1 Quick Usage Guide

Following is a quick guide to using diagnostics with links to the appropriate sections.

- Enable diagnostics from the Diagnostics Manager: **HERE (Enable Information Diagnostics:)**
- Specify the destination of the diagnostics: **HERE (Specify Diagnostic Destination:)**
- Select the diagnostic categories you wish to print: **HERE (Section 1.3)**
- Filter the diagnostics by Object, Slot, Timestep, Rule, or Account: **HERE (Section 1.3.2)**
- Apply the changes in both the Diagnostic Configuration dialog and the Diagnostics Manager dialogs:
- Run the model and view the diagnostics in the specified file or Diagnostics Output: **HERE (Section 1.4)**
- Additional information on debugging RPL logic using diagnostics or other tools can be found **HERE (RPLDebugging.pdf, Section 3)**.

1.2 Diagnostics Manager

The Diagnostics Manager is used to enable informational diagnostics. Informational diagnostics are any messages which, although helpful in debugging, are not indicators of problems which would stop a modeling run. Informational diagnostics should not be confused with RiverWare warnings and errors. Whereas informational diagnostics may be filtered by context and turned off altogether, warnings and errors may not. Warnings and errors are always displayed in the **Diagnostics Output window** and always force this window to come to the front of other open windows.

The diagnostics Manager can be accessed from the:

- **Workspace ➡ Utilities ➡ Diagnostics Manager**
- **Run Control ➡ View ➡ Diagnostics Manager**
- **Diagnostics Output ➡ Settings ➡ Show Diagnostics Manager Dialog**

RiverWare is initially started with the diagnostic capability turned **OFF** primarily for efficiency. Posting, filtering, and outputting unnecessary diagnostic messages may take up a significant portion of the run time. This should be kept in mind when activating diagnostic categories and filters. An excessive amount of diagnostic messages produced at run time may considerably slow down the execution.

The following should be done to enable diagnostics:
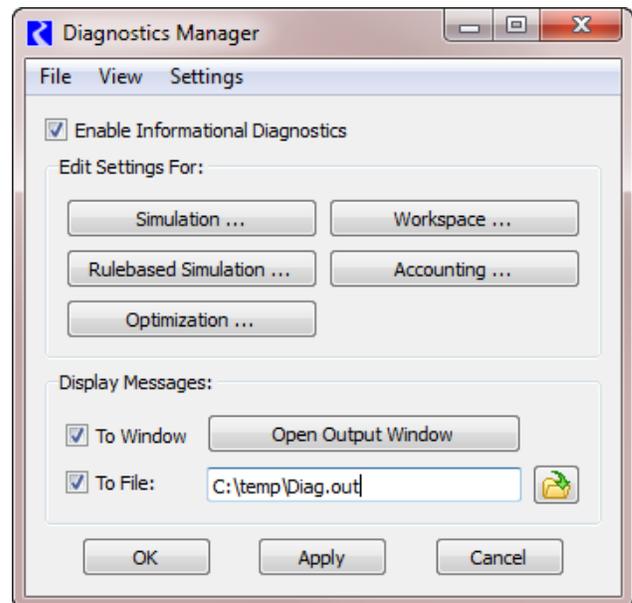
Enable Information Diagnostics: check the toggle next to the **Enable Informational Diagnostics** box. Do this only if the box is not already checked. (The diagnostics are turned on when the box next to **Enable Informational Diagnostics** is checked).

Specify Diagnostic Destination: The destination of output messages is set by checking the **To Window** and/or **To File:** boxes. In the case of file output, the file name and path must also be specified in the text field or through the file chooser invoked by clicking on the **Select File** button 🗁 . If neither output format is specified, diagnostic messages will be printed in the standard output window. In Windows, the standard output window defaults to a DOS command prompt. Do not close this command prompt window (minimizing is ok) as it will also close RiverWare with no warning. In UNIX, the standard output window is the terminal window from which RiverWare was originally launched.

The **Diagnostics Manager** dialog window also contains three buttons at the bottom: **OK**, **Apply**, and **Cancel**. Changes to the dialog must be followed by pressing either the **Apply** or the **OK** buttons in order for the changes to take effect. The **OK** button closes the dialog in addition to applying any new settings. The **Cancel** button may be pressed to cancel any changes made to the dialog since the last **Apply** or **OK**, and close the dialog.

Diagnostics are divided into categories which appear in the upper portion of the Diagnostics Manager. Each category is a button which opens up the **Diagnostics Configuration** dialog for that category.:

- Simulation **HERE (Section 2)**
- Rulebased Simulation **HERE (Section 3)**
- Optimization **HERE (Section 4)**
- Workspace **HERE (Section 5)**
- Accounting **HERE (Section 6)**

The first three categories are associated with the controllers of the same name. Accounting diagnostics are enabled only if the Accounting is enabled. For Simulation and Rulebased Simulation, diagnostic messages are generated and filtered **during** a run according to the settings specified for the currently active controller only. The Workspace category, however, is independent of the current controller. It is active **before** and **after** a model run. For example, diagnostic for expression slot that are configured to execute at the end of a timestep must be set up in either the Simulation or Rulebased Simulation category. Diagnostics for expression slots that are evaluated manually must be configured from the workspace menu.

1.2.1 Loading and Saving Diagnostic Configurations

Once the you has configured the diagnostics settings, these settings can be saved to a file and loaded at a later time. This allows you to configure and save various combinations of diagnostics configurations and context filters. Settings are loaded and saved by selecting **File ➥ Load Settings** or **File ➥ Save Settings** in the **Diagnostics Manager** dialog.

1.3 Diagnostic Configuration Dialogs

Each of the diagnostics configuration dialogs contains a group of diagnostics to turn on or off and a set of context filters. The available groups are shown below the title **Show diagnostics for:** in the upper left section of the dialog window. Each of these diagnostics groups can be toggled on or off by clicking on the box located to the left of the group name. Multiple groups can be turned on at once. Some diagnostics groups are only visible by toggling a tree view triangle.

The mechanism which allows you to narrow down the number of diagnostics that will appear is called context. The various contexts are: object, slot, timestep, and rule. Not all of these contexts are valid for every configurations dialog. For example, generating diagnostics about the dispatching of objects is independent of any Slot context. The **Simulation Diagnostics Configuration** dialog only allows filtering of messages by Object, Slot, and/or Timestep. Selecting a limited set of a particular context, restricts diagnostic generation to that set. For instance, the **<ALL OBJECTS>** within the **Show O diagnostics for:** field means that object-related diagnostics will be generated for every object that exists on the RiverWare workspace. If, however, a limited set of objects is listed within the field, then object-related diagnostics will only be generated for that select set of objects.

For any average size model, enabling this many diagnostics is dangerous. The time spent posting, processing, and displaying diagnostic messages may "freeze" model execution. For a small model, the effect is limited to a noticeable slowdown. In general, diagnostics should never be enabled without considerable context filtering. The proper use of context filtering is discussed **HERE (Section 1.3.2)**.

Following is a description of the Diagnostics Groups and the Context filtering available in the Diagnostics Manager.

1.3.1 Diagnostics Groups

Diagnostics Groups are the broad categories of diagnostics which can be generated. They include such things as object dispatching information, rules firing information, rules agenda information, slot value
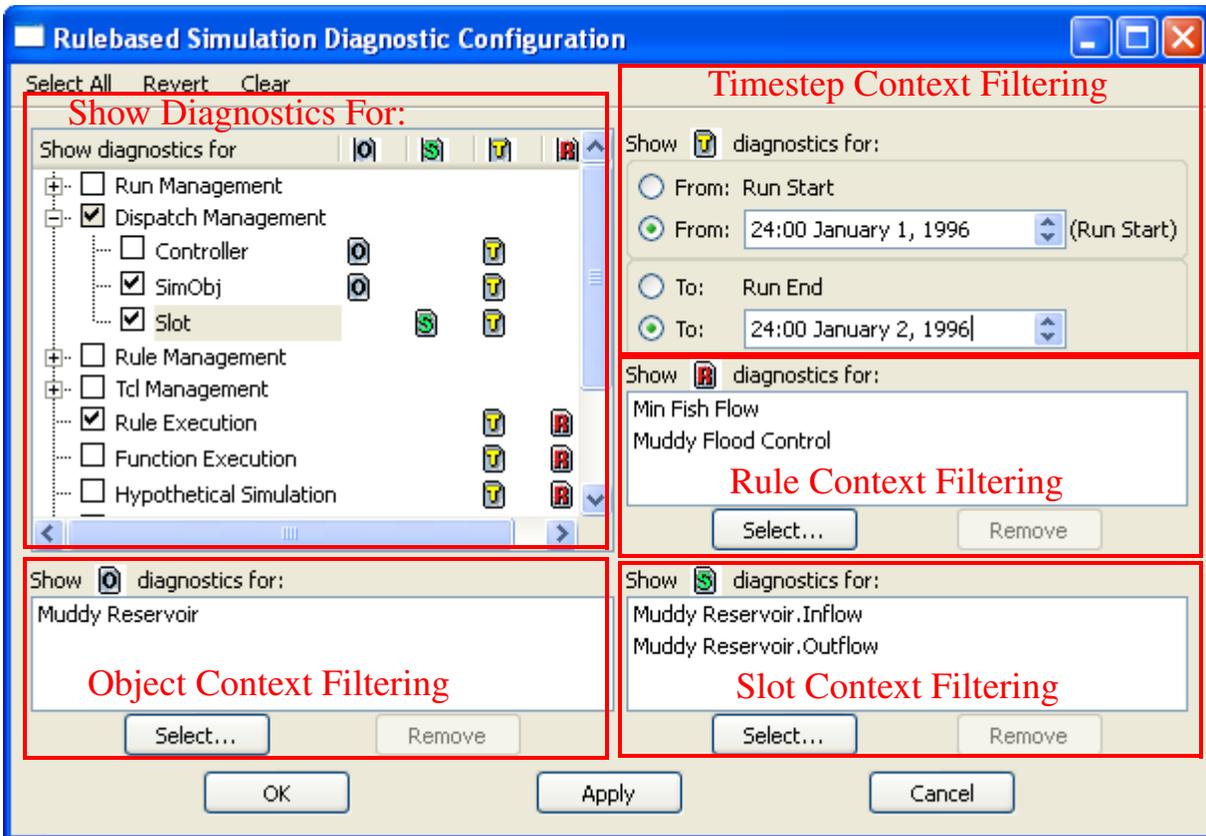
information, etc. A complete list of the types of messages issued by each diagnostics group is presented in subsequent sections.

> DIAGNOSTICS GROUP
>
> A diagnostics group is a category of diagnostic messages which pertain to similar events within a run.

Diagnostics groups are selected in the upper left portion of the dialog, in the window below the **Show diagnostics for:** label. The diagnostics groups are turned on by clicking in the box to the left of the group name. When a diagnostics group is turned on, a check mark is shown to the left of the group's name. Some of the items in the list are actually groups of diagnostics groups. These "super-groups" are purely for organization and generate no diagnostics. The sub-groups within a super-group may be viewed by clicking on the tree-view plus symbol to the left of the name. This action will be shown in this manual as "**Super-Group ➡ Sub-Group**." When any of the diagnostic sub-groups in a super-group are turned on, a check mark is shown to the left of the super-group name, in addition to the green check mark to the left of each sub-group name. Any number of diagnostics groups may be turned on at the same time. When you click a Super-Group check mark, it uses a "tri-state": All On, All off, previous sub-group selection.

1.3.2 Context Filtering

Every diagnostic message is generated within a specific context. Contexts are details about the source of a diagnostic message. The contexts for a message may include the Object, Slot, Timestep, and/or Rule within which it originated.

> DIAGNOSTIC CONTEXT
>
> Diagnostic context identifies the Object, Slot, Timestep, and/or Rule from within which a message is generated.

All messages which are part of the same diagnostics group will have the same types of contexts. The contexts which are part of each diagnostics group are indicated by a symbol(s) to the right of the diagnostics group name. An icon exists to represent each context type as follows:

	Object context
	Slot context
	Timestep context
	Rule Context
	Account Context
	Goal Context

Each of the icons contains a letter abbreviation of the type of context, **O** for object, **S** for slot, **T** for timestep, **R** for rule, **A** for account, and **G** for optimization goal. Not all **Diagnostics Configuration** dialogs contain the same contexts. The table below summarizes the contexts available to all of the **Diagnostics Configuration** dialogs.

DIAGNOSTICS CONFIGURATION DIALOG	AVAILABLE CONTEXTS
Simulation	Object, Slot, Timestep
Rulebased Simulation	Object, Slot, Timestep, Rules
Optimization	Object, Timestep, Goal
Workspace	Object, Slot
Accounting	Object, Slot, Timestep, Account

Exactly how many icons are displayed depends on the Diagnostics Configuration dialog invoked, as indicated in the table above.

Context filters are important for reducing the number of messages produced by a diagnostics group. Context filters allow you to specify the only source(s) from which a diagnostic message will be displayed. Any messages originating from a different source are ignored.

Most diagnostic groups are affected by context(s). To the right of every diagnostic group listed in the **Show diagnostics for:** field, there are zero or more context icons indicating whether or not those particular types of diagnostics are generated within the depicted context types. For example, the posting of diagnostic group **Dispatch Management ➥ Slot** messages is affected by Slot and Timestep context. Both the Slot and Timestep context icons appear to the right of the listed diagnostic group. In this way, you can determine whether selecting a limited set of items within a context type will affect the frequency of messages of a particular diagnostic group. Choosing a limited set of slot names, for example, only affect the frequency of diagnostics groups depicting the slot context icon **S**.

The summary context icons above the diagnostic group list box indicate whether or not diagnostic groups affected by the specific context are turned on. If none of the diagnostic groups are turned on, then diagnostics are not generated from any of the existing contexts. As a result, each of the summary context icons above the list box is turned OFF, and each has a red line through it. As soon as a single diagnostic group is turned ON and is related to a specific context, the corresponding summary context icon is turned ON (the red line is removed.)

Example

Turning on the **Dispatch Management ➥ Controller** diagnostics group for a model with 60 objects and 5 years of monthly timesteps, would generate at least 3600 messages. This is too many messages to look through to verify that a particular object dispatched on a particular timestep. Because the diagnostics group has both object and time context (as indicated by the symbols to the right of its name), we can set context filters to ignore all messages which did not originate from the desired object and timestep. This reduces the number of diagnostic messages to look through from approximately 3600 to 1.

Object List: The object list in the **Simulation Diagnostic Configuration** dialog contains a list of the objects for which applicable diagnostics will be generated. To add or delete members of the list, click the **Select...** button under the **Show O diagnostics for:** field. This invokes an **Object Selector** dialog. Highlighted objects within the list may be removed directly by pressing the **Remove** button.

Timestep Selection: In the upper right corner of the Diagnostics Configuration dialog, there is a **Show T diagnostics for:** field, which allows you to input the times within which diagnostics (affected by timestep context) will be generated. By default, diagnostics are posted from the start of the run to the end of the run. This can be overridden by selecting the toggle buttons to the left of the **From** and **To** fields. Times are input in standard HH:MM MM/DD/YY format, e.g., 6:00 02/06/97. In addition, there is a toggle to specify the Run Start and Run End. The date/time spinners to the right of text fields can also be used to select the desired times.

Slot List: The Slots list is modified in the same manner as the Objects list, with one added feature; Slots selected in the Slot Selector dialog

Rule List: In the **Rulebased Simulation Diagnostic Configuration** dialog, a box is displayed to allow you to select rules for which diagnostics should be displayed. Click **HERE (Section 3.2)** for more information.

Accounts List: In the **Accounting Diagnostics Configuration** dialog, a box is displayed to allow you to select accounts for which diagnostics should be displayed.

Goal List: In the **Optimization Diagnostic Configuration** dialog, a box is displayed to allow you to select goals for which diagnostics should be displayed.

1.3.3 Select All, Revert and Clear

These three menu options are available within each of the diagnostics configuration dialogs, depending on whether or not they exist in the specific Configuration dialog. Each of these options can be applied to diagnostics groups, the object list, the slot list, the rule list, and the timestep text fields.

Applying the **Select All** feature has the following effects:

|	
Diagnostics Groups	All turned to ON
Objects	ALL listed in listbox
Slots	ALL listed in listbox
Rules	ALL listed in listbox
Goals	ALL listed in listbox
Times	Set to StartTime and EndTime

The **Revert** feature reverts the appropriate items back to their state prior to your modifications. Every time the **Apply** button is pressed, your changes are recorded. Selecting the **Revert** command will revert back to the settings which were last recorded.

Applying the **Clear** feature has the following effects:

|	
Diagnostics Groups	All turned to OFF
Objects	NONE listed in listbox
Slots	NONE listed in listbox

1.4 Diagnostic Output Window

The **Diagnostics Output Window** displays the diagnostics during a RiverWare run. Error and Warning messages are also displayed in this window, regardless of user settings, and automatically force the window to the top of any other windows. The object and timestep for which a message is generated are listed under the **Context** bar. The message containing the diagnostic information is listed under the **Diagnostic Message** bar on the same line. In the top right corner, there is a box with the **Auto Scroll** label to the right. When this box is selected (i.e. the box is dark or checked), the window will continually scroll down as messages are output. In a large model where many diagnostics are generated, the continuous scrolling can add significantly to the run time. In these cases, the **Auto Scroll** can be turned off by clicking in the box.
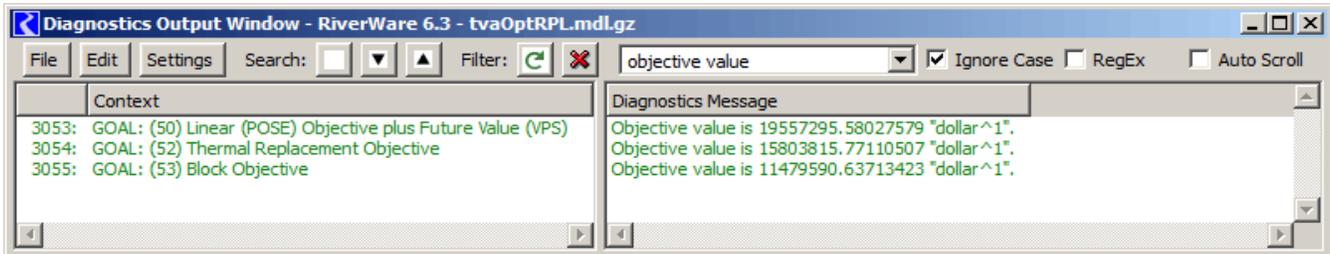
You can scroll vertically and horizontally in both the context and message fields. In addition to scrolling in the fields, you may wish to resize the entire **Diagnostics Output Window**. This is useful when many long messages and/or contexts need to be read. To move the divider between the fields, position the mouse over the space between the heading bars, press the left mouse button, and drag the divider to the desired width.

Long diagnostic messages are shown as a wrapped tooltip when you hover over a message.
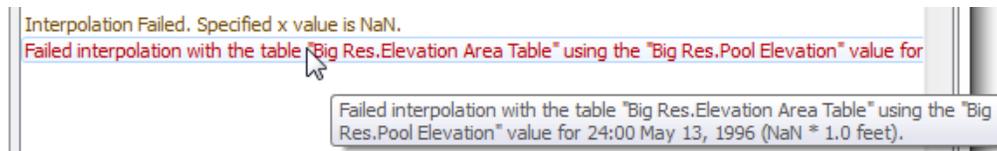
In the **Diagnostics Output Window,** the following right-click context menu operations are available:

- **Open Context Object**: Open the dialog for the listed Object, Account, Slot, RPL Set, RPL Rule, RPL Goal, or RPL Group.

- **Copy Context Timestep**: Copy the datetime to the clipboard
- **Global Scroll to Context Timestep**: Global scroll to the datetime
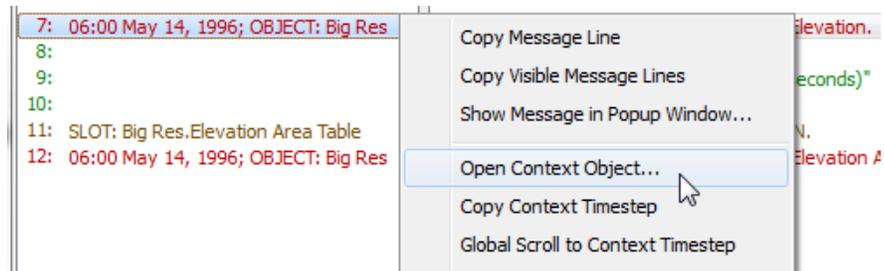
If you wish to read a diagnostic in a wrapped window, right click on the line of text and choose **Show Message in Popup Window.**
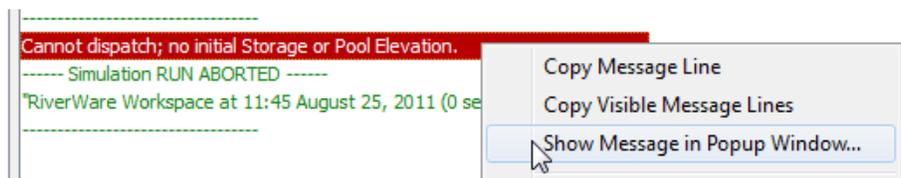
This opens a dialog that shows the full diagnostics text, including the context, in a wrapped window. If you highlight the name of an Object, Slot, Account, RPL item (Set, Group, Rule, Function), you can click the **Open Selected Named Item** menu to open the corresponding dialog.
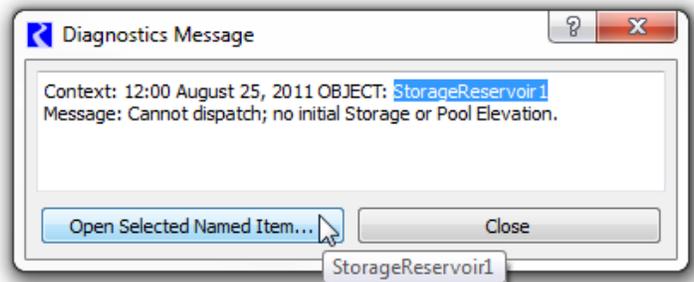
1.4.1 Message Search and Filter Controls

There are two approaches used to find messages: **Search** and **Filter**. Search to look for a specific text string row by row. Filter to show all rows that match the text string. (You can also search on filtered results) The two approaches share the text entry field. Following are descriptions of the controls and options:
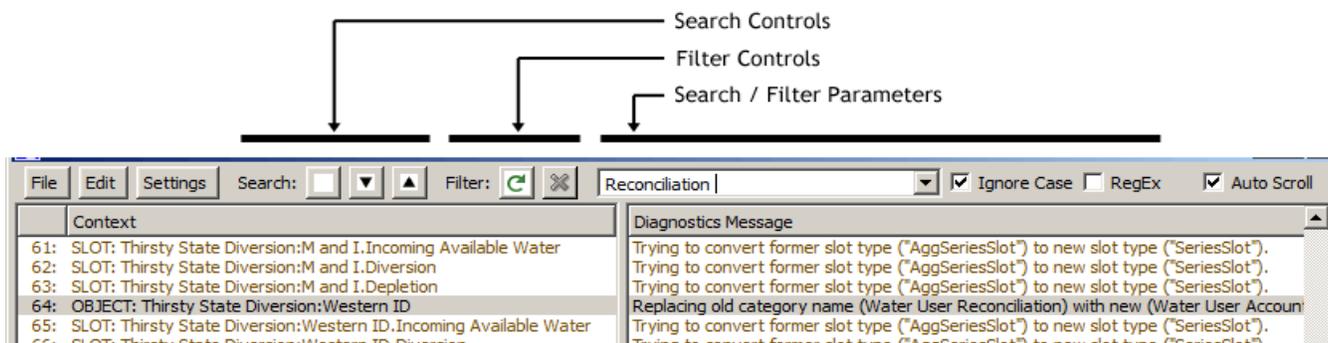
Search Controls: The **Search** controls include:

- **Message Type** (color coded) Drop-Down menu. Filter the search by the type of message displayed, i.e. by color. Click on the white box just to the right of the Search field. This opens a pull down menu as shown in the following screenshot. Choose an item and any subsequent searches will be filtered to only include messages of that type. If you just want to see the next message of a certain type, don't enter anything in the search box; a search will jump to the next message of the indicated color/type.
- **Search Forward** button (down) ▼
- **Search Backward** button (up) ▲
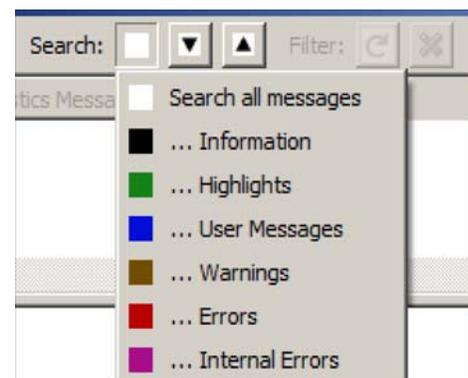
> **Note:** TIP: The "Add entered text as User Message" operation in the Edit menu can be used to effectively append a "bookmark" to the end of the message list. That text can later be recalled from the history menu of the entry field, and used to scroll to the various instances of that bookmark within the message list using the Search functions.

Filter Controls: Filtering allows you to show only those lines that match the specified text. The Filter controls include these two buttons.:

1. The **Apply Filter** button ⟳ applies the entered text as the active filter pattern. Message rows not matching the pattern are hidden. This button is disabled when the search / filter parameters already match the currently active filter (or when the entered text is empty).

2. The **Clear Filter** ✖ button clears filtering -- causing all message rows which had been hidden to become visible again. This button is enabled only when filtering is active.

The only direct indication that messages are being filtered is the Clear Filter (red "X") button being enabled and discontinuous line numbers in the message list.

When applying or clearing a filter, the view is re-scrolled to the currently selected item (when that item remains non-hidden). So, it is possible to use filtering to locate a diagnostic message (and "click" that message), and then view that message in the context of all generated messages by turning filtering off.

Search/Filter Parameters: Enter your desired search/filter text pattern in the text box. This is an editable "combo box" in order to support a history of recently applied patterns as shown to the right:

When typing in the entry field, hitting Enter is equivalent to clicking the "Search Forward" button, and hitting Shift-Enter is equivalent to clicking the "Search Backward" button.

The two following options apply to both the **Search** and **Filter** functions.

- The **Ignore Case** checkbox causes upper/lower case differed to be ignored when comparing the pattern text to message text.
- The **RegEx** checkbox changes the interpretation of the pattern text from "Wild Card" to "Regular Expression" syntax. (See the following section).

Note: The search functions skip over messages which have been hidden by filtering.

Search and Filter Pattern Matching Technical Details: A given message line matches the pattern if the line text contains the pattern. (This is different from the pattern matching the entire line of text). Application of the pattern match concatenates the "context" and "message" parts of the message row, rather than independently looking into those two parts. This makes possible, for example, a wildcarded
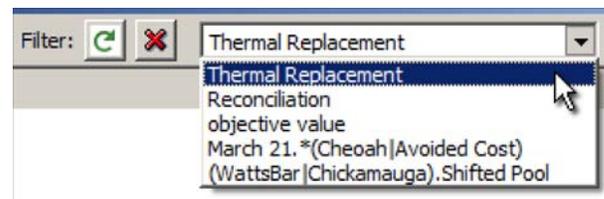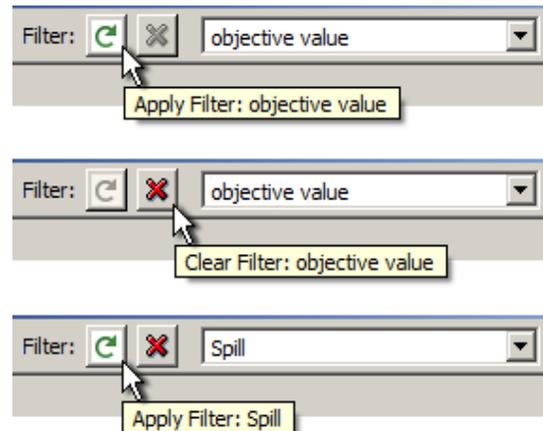
expression which matches pieces from both the context and message parts of the message line, as illustrated here: `February 23, 1996*Propagate*returnFlow`
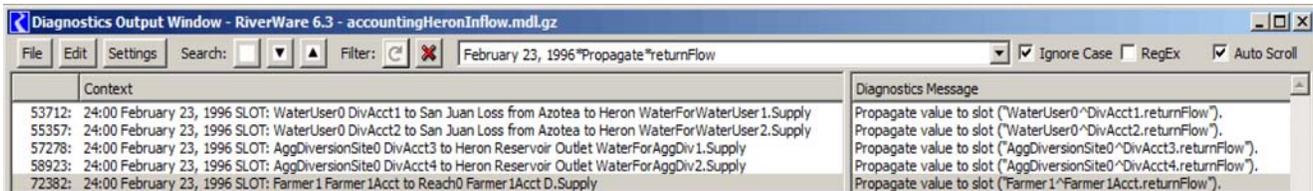
The following is the definition for "Wild Card" pattern matching (i.e. not RegEx):

- Any character represents itself apart from those mentioned below.
- "?" (question mark) matches any single character. It is the same as "." (period) in regular expressions.
- "*" (asterisk) matches zero or more of any characters. It is the same as ".*" (period asterisk) in regular expressions.
- Sets of characters can be represented in square brackets [...].
- A back-slash ('\') "escapes" the wildcard characters. It is possible to match a question mark, asterisk, or square bracket by preceding ("escaping") those characters with a back-slash ("\").

Regular Expression ("RegEx", "regex" or "regexp") pattern syntax is an advanced, potentially complex pattern matching format developed originally with Unix-based tools. It's not described further in this document but is widely referenced online.

1.4.2 File Menu

The **File** menu, contains:

- **Save Diagnostic Messages to File:** Save all diagnostics in the window to a text file
- **Show Diagnostics Messages in Stand Alone-Window**: When the diagnostics output is docked in an SCT, this menu option allows you to show the messages in a stand-alone window
- **Show Workspace**: bring the workspace to the front
- **Close Diagnostic Output Window**: close the window

1.4.3 Edit Menu

The **Edit** menu contains options to:

- **Copy Message Line**: copy a single line to the clipboard
- **Copy Visible Message Lines**: copy visible lines to the clipboard
- **Copy All Message Lines**: copy all the diagnostics to the clipboard
- **Add entered text as User Message**: This option takes any text that is in the Search text box and creates a new message line. This is useful in debugging if you want to place a marker
- A **Clear Messages** command clears the existing diagnostics from the scrollable listbox. During the course of running RiverWare with diagnostics enabled, the number of diagnostics can grow very large (especially if they are being generated for **<ALL OBJECTS>** and **<ALL SLOTS>** context types, and all

Times). As a result, **Clear Messages** whenever possible to decrease the amount of memory used by RiverWare. Diagnostics are appended to the bottom of the window and are not deleted from the window until you invoke the Clear command. As a result, if you are making multiple runs, the diagnostics from the first run remain visible at the top of the window. When looking at diagnostics, make sure to always start at the bottom to ensure you are viewing diagnostics from the current run.

1.4.4 Settings Menu

The **Settings** menu contains

- **Diagnostics Manager...:** Open the diagnostics manager.
- **Show Color Legend...**: Display the meaning of diagnostic message colors.
- **Show Column Headers:** Toggle to either show or hide the column headings.
- **Show Line Numbers:** Toggle to either show or hide line numbers.
- **Show Filter Statistics:** Toggle to either show or hide a statistics panel below the message list. This indicates the total number of diagnostics messages, and how many are shown and hidden due to message filtering.
- **Show Horizontal Scroll Bars:** Toggle to either show or hide scroll bars
- **Search/Filter: Ignore Case**: Toggle to ignore case in the search/filter box.
- **Search/Filter**: **Use Regular Expressions:** Toggle to use Regular Expressions in the search/filter box.
- **Automatic Width Adjustment**: The "Context" portion of the message list (on the left side) is given the full width necessarily to display its content for all non-hidden messages.
- **Extra Width Fix:** This option is intended to correctly measure the text content on certain Windows machines. When this option is on, automatically adjusted widths are increased by about 12%

Simulation Diagnostics

2. Simulation Diagnostics

2.1 Simulation Diagnostic Groups

Below is a sample Simulation Diagnostic Configuration Dialog. Available filters include Object, Timestep, and Slots.

The table below indicates the kinds of messages which are displayed when a Diagnostic Group is selected. Examples are shown in Helvetica Bold font.

RiverWare Technical Documentation: Diagnostics
Revised: 7/17/18

Super Group	Diagnostics Group	Type of diagnostics issued
Run Management:		
	Controller	Beginning/End of Run/Timestep for each time. Examples: • **Initializing model run.** • **Beginning of Run.** • **Begin Timestep.** • **Execute timestep.** • **End of run.**
	SimObj	Beginning/End of Run/Timestep for each Object and time. Examples: • **Clear state.** • **Propagating user input.** • **Beginning of run.** • **Begin timestep.** • **End timestep.** • **End of run.**
	Slot	Beginning of Run for each Slot and time. Examples: • **Clear state.** • **Propagating user input.**
Dispatch Management:		
	Controller	Beginning of a dispatch for each object and time. Example: • **Executing dispatch method (solveMB_givenInflow-Outflow) at Priority 7.**

Super Group	Diagnostics Group	Type of diagnostics issued

	SimObj	Dispatch readiness for each object and time.
		
		Examples:
		
		• **Slot ("Inflow") added to known set.**
		• **Ready for dispatch method ("solveNROutflow").**
		• **Slot ("Inflow") is already in known set.**
		• **Re-dispatching using slot priorities.**
		  Priority 3R:Inflow
		  Priority 7:Outflow
		  Priority 7:Total DIversion
		  Priority 7:Total Return Flow
		  Priority 7:Total Local Inflow
		  Conflicting slot values:("Inflow" "Outflow" "Total Diversion" "Total Return Flow" "Total Local Inflow").

	Slot	Value assignments and propagation for each slot and time. Examples: • **Set value = 453.003351 *1000.0 acre-feet/month = 208.621343 *1.0cms.** • **Propagate value to slot ("Total Available Water").**
Independent Groups:		
	User Methods	Execution of User Methods for each object and time. Example: • **Executing user method ("DailyEvaporationCalc").**
	Interpolation	TableSlot table interpolation lookups for each slot. Examples: • **Table Interpolation 2 (3D) Specified z value (960.6579) falls between row ranges (556-572) and (573-589). (1->2) rows (556-572) x=195.804788, y=198.503173 (1->2) rows (573-589) x=195.804788, y=198.585469 (fixed 0 at 960.657920, 1->2) (in=195.804788) (out=198.579323 slope=.656000 inter-cept=70.131382)**
	Client Server	Provides information on the Client Server interaction. This includes the RiverWare - DSS , RiverWare - HDB, and RiverWare - MODFLOW connections.
	Expr. Slot Execution	Expression slot execution are printed for expression slots that are configured to evaluate during a run (evaluation time of beginning or end of timestep or beginning or end of run). Example: • **Expression evaluated to 624.00 [1000 acre-ft]**

| Super Group	Diagnostics Group	Type of diagnostics issued
	Expr. Slot Function Execution	Evaluation of RPL functions called by expression slots evaluated during a run. Diagnostics messages are printed for function execution (both user defined and pre-defined). "Before Execution" function diagnostics show the function being called and the arguments passed into the function. "After Execution" function diagnostics show the value that the function evaluated to and whether or not the value was constrained.
		NOTE: Function diagnostics also need to be enabled on the functions themselves. This can be done through the **Function Editor** dialog (for enabling diagnostics on individual functions) or by selecting **Set ➡ Function Diagnostics** from the main **Set Editor** (for enabling or disabling function diagnostics for all functions in the set). The Expression Slot Set editor is accessed from the Workspace **Policy ➡ Open Expression Slot RPL Set**. Click **HERE (Section 3.3.3)** for more information on configuring the diagnostics or **HERE (Slots.pdf, Section 4.6)** for more information on expression slots.
		Example:
		• **SumFlowsToVolume($"Mountain Storage.Inflow", @"Start Timestep", @"t")**
		• **"SumFlowsToVolume" evaluated to: 274839.74 ["m3"].**
	Init. Rules Rule Execution	Initialization Rules rule execution diagnostics are printed for each rule.
		For a Simulation run, there is no filtering based on rule name. When enabled, diagnostics are shown for all Initialization rules.
		Examples:
		• **Executing rule (#3)** **Assignment initiated (the left-hand side is "Mohave.Storage"[])** **Evaluation of Assignment statement successful; will attempt assignment** **Mohave.Storage[July, 1998] = 1590800.00 [1.00 acre-ft]** **Rule successfully finished**
		• **Early termination: NaN encountered.** **The problem was encountered at the following location within the** **expression: "FC Violation"().**
		• **RHS of assignment evaluated to NaN.**

Super Group	Diagnostics Group	Type of diagnostics issued
	Init. Rules Function Execution	Diagnostics messages for function execution (both user defined and pre-defined) from initialization rules. For a Simulation run, there is no filtering based on rule name. When enabled, diagnostics are shown for all Initialization rules. "Before Execution" function diagnostics show the function being called and the arguments passed into the function. "After Execution" function diagnostics show the value that the function evaluated to and whether or not the value was constrained. NOTE: Function diagnostics also need to be enabled on the functions themselves. This can be done through the **Function Editor** dialog (for enabling diagnostics on individual functions) or by selecting **Ruleset ➡ Ruleset Function Diagnostics** from the main **Ruleset Editor** (for enabling or disabling function diagnostics for all functions in the ruleset). Click **HERE (Section 3.3.3)** for more information. Examples: • **GetMaxReleaseGivenInflow (% "WattsBar", 0.634 ["1000 * cms"], @"Current Timestep")** • **"GetMaxReleaseGivenInflow" evaluated to: 1299.413 ["cms"]** • **"GetMaxReleaseGivenInflow" constrained to: 1200.00 ["cms"]**
	Init. Rules Print Statements	User defined messages for each initialization rule. For a Simulation run, there is no filtering based on rule name. When enabled, diagnostics are shown for all Initialization rules. Example: • **Entering rule curve routine for January - July**

Rulebased Simulation Diagnostics

3.   Rulebased Simulation Diagnostics

The importance of diagnostics in debugging Rulebased Simulation models cannot be overstated. Diagnostics allow the user to examine rule firing, rule success/failure, slot assignments, and dispatching in the chronological order in which these events take place. It is the only way to verify that the intended interaction between rules and object dispatching is being carried out exactly as expected. Becoming familiar with configuring and reading diagnostics will save you hours and hours of speculation when your model does not behave the way you planned. Additional information on debugging Rules and other RPL logic using **diagnostics** can be found **HERE (RPLDebugging.pdf, Section 3)**. A more general discussion on debugging RPL logic can be found **HERE (RPLDebugging.pdf, Section 1)**.

The **Rulebased Simulation Diagnostics Configuration** dialog allows you to select from fifteen diagnostics groups and four context filters which are relevant during a Rulebased Simulation run. This dialog's settings are only valid when the Rulebased Simulation controller is selected. It is accessed from the Rulebased Simulation button on the Diagnostics Manager.

3.1 Rulebased Simulation Diagnostic Groups

The table below indicates the kinds of messages which are displayed when a Diagnostic Group is selected. Examples are shown in Helvetica Bold font.

Super Group	Diagnostics Group	Type of diagnostics issued
Run Management:		
	Controller	Beginning/End of Run/Timestep for each time. Examples: • **Initializing model run.** • **Beginning of Run.** • **Begin Timestep.** • **Execute timestep.** • **End of run.**
	SimObj	Beginning/End of Run/Timestep for each Object and time. Examples: • **Clear state.** • **Propagating user input.** • **Beginning of run.** • **Begin timestep.** • **End timestep.** • **End of run.**
	Slot	Beginning of Run for each Slot and time. Examples: • **Clear state.** • **Propagating user input.**
Dispatch Management:		
	Controller	Beginning of a dispatch for each object and time. Example: • **Executing dispatch method (solveMB_givenInflow-Outflow) at Priority 7.**

Super Group	Diagnostics Group	Type of diagnostics issued

	SimObj	Dispatch readiness for each object and time.
		
		Examples:
		
		• **Slot ("Inflow") added to known set.**
		• **Ready for dispatch method ("solveNROutflow").**
		• **Slot ("Inflow") is already in known set.**
		• **Re-dispatching using slot priorities.**
		  Priority 3R:Inflow
		  Priority 7:Outflow
		  Priority 7:Total DIversion
		  Priority 7:Total Return Flow
		  Priority 7:Total Local Inflow
		  Conflicting slot values:("Inflow" "Outflow" "Total Diversion" "Total Return Flow" "Total Local Inflow").

	Slot	Value assignments and propagation for each slot and time. Examples: • **Set value =453.003351 *1000.0 acre-feet/month =208.621343 *1.0cms.** • **Propagate value to slot ("Total Available Water").**
Rule Management:		
	Dependencies	Dependencies for each rule and time. Example: • **After execution, rule #4 has the following dependencies: Mead.Storage, Mead.Inflow.**
	Agenda	Additions and status of the agenda for each rule and time. Examples: • **The following rules are currently on the agenda: 3, 4, 8.** • **Changing dependency (Mead.Outflow) caused rule #3 to be put on the agenda.**
	Assignments	Value assignments for each rule, time and slot. Example: • **Result of sending values to controller:** **- Priorities permitted setting of Mead.Storage[24:00 January 31, 1993] to 676194743.34m3**
Independent Groups:		

Super Group	Diagnostics Group	Type of diagnostics issued

	Rule Execution	Rule execution details for each rule, time and slot.
		
		Examples:
		
		• **Executing rule (#3)**
		  Assignment initiated (the left-hand side is "Mohave.Storage"[])
		  Evaluation of Assignment statement successful; will attempt assignment
		  Mohave.Storage[July, 1998] = 1590800.00 [1.00 acre-ft]
		  Rule successfully finished
		• **Early termination: NaN encountered.**
		  The problem was encountered at the following location within the
		  expression: "FC Space Violation"().
		• **RHS of assignment evaluated to NaN.**
		• **Unable to set slot (Outflow at 00:00 February 1, 1999)**
		  Rule's priority (4) junior to slot's (2).

	Function Execution	Diagnostics messages for function execution (both user defined and pre-defined). "Before Execution" function diagnostics show the function being called and the arguments passed into the function. "After Execution" function diagnostics show the value that the function evaluated to and whether or not the value was constrained. NOTE: Function diagnostics also need to be enabled on the functions themselves. This can be done through the **Function Editor** dialog (for enabling diagnostics on individual functions) or by selecting **Ruleset** ➥ **Ruleset Function Diagnostics** from the main **Ruleset Editor** (for enabling or disabling function diagnostics for all functions in the ruleset). Click **HERE (Section 3.3.3)** for more information. Examples: • **GetMaxReleaseGivenInflow (% "WattsBar", 0.634 ["1000 * cms"],** **@"Current Timestep")** • **"GetMaxReleaseGivenInflow" evaluated to: 1299.413 ["cms"]** • **"GetMaxReleaseGivenInflow" constrained to: 1200.00 ["cms"]**

Super Group	Diagnostics Group	Type of diagnostics issued
	Hypothetical Simulation	Diagnostics messages for hypothetical simulation. This includes information about the start and end of hypothetical simulation, the iteration information (if using a target hypothetical simulation function), and any information regarding the hypothetical simulation solution. Examples: • **HypSim: Hypothetical limit simulation starting for specified SubBasin.** • **Hypothetical simulation occurring within the time range [12:00 November 23, 1995 - 00:00 December 5, 1995].** • **HypSim: For iteration 1, control slot interval is [0.000000 "cms", 75.000000 "cms"], range of interval is [1768.684936 "m", 1763.270439 "m"].** • **HypSim: For target value 1768.297200 "m", result is control value 19.628906 "cms", leading to the simulated target value 1768.296428 "m" (error = 0.000772 "m").** • **HypSim: Hypothetical limit simulation ending for specified SubBasin.**
	Print Statements	User defined messages within rules for each rule and time. Example: • **Entering rule curve routine for January - July**
	User Methods	Execution of User Methods for each object and time. Example: • **Executing user method ("DailyEvaporationCalc").**
	Interpolation	TableSlot table interpolation lookups for each slot. Examples: • **Table Interpolation 2 (3D) Specified z value (960.6579) falls between row ranges (556-572) and (573-589). (1->2) rows (556-572) x=195.804788, y=198.503173 (1->2) rows (573-589) x=195.804788, y=198.585469 (fixed 0 at 960.657920, 1->2) (in=195.804788) (out=198.579323 slope=.656000 intercept=70.131382)**
	Expr. Slot Execution	Expression slot execution diagnostics are printed for expression slots that are configured to evaluate during a run (evaluation time of beginning or end of timestep, or beginning or end of run). Example: • **Expression evaluated to 624.00 [1000 acre-ft]**

Super Group	Diagnostics Group	Type of diagnostics issued
	Expr. Slot Function Execution	Evaluation of RPL functions called by expression slots evaluated during a run. Diagnostics messages are printed for function execution (both user defined and pre-defined). "Before Execution" function diagnostics show the function being called and the arguments passed into the function. "After Execution" function diagnostics show the value that the function evaluated to and whether or not the value was constrained. NOTE: Function diagnostics also need to be enabled on the functions themselves. This can be done through the **Function Editor** dialog (for enabling diagnostics on individual functions) or by selecting **Set ➥ Function Diagnostics** from the main **Set Editor** (for enabling or disabling function diagnostics for all functions in the set). The Expression Slot Set editor is accessed from the Workspace **Policy ➥ Open Expression Slot RPL Set**. Click **HERE (Section 3.3.3)** for more information on configuring the diagnostics or **HERE (Slots.pdf, Section 4.6)** for more information on expression slots. Example: • **SumFlowsToVolume($"Mountain Storage.Inflow", @"Start Timestep", @"t")** • **"SumFlowsToVolume" evaluated to: 274839.74 ["m3"].**

	Init. Rules Rule Execution	Initialization Rules rule execution diagnostics are printed for each rule.
		
		Examples:
		• **Executing rule (#3)**
		  Assignment initiated (the left-hand side is "Mohave.Storage"[])
		  Evaluation of Assignment statement successful; will attempt assignment
		  Mohave.Storage[July, 1998] = 1590800.00 [1.00 acre-ft]
		  Rule successfully finished
		• **Early termination: NaN encountered.**
		  The problem was encountered at the following location within the
		  expression: "FC Violation"().
		• **RHS of assignment evaluated to NaN.**

Super Group	Diagnostics Group	Type of diagnostics issued
	Init. Rules Function Execution	Diagnostics messages for function execution (both user defined and pre-defined) from initialization rules. "Before Execution" function diagnostics show the function being called and the arguments passed into the function. "After Execution" function diagnostics show the value that the function evaluated to and whether or not the value was constrained. NOTE: Function diagnostics also need to be enabled on the functions themselves. This can be done through the **Function Editor** dialog (for enabling diagnostics on individual functions) or by selecting **Ruleset ➡ Ruleset Function Diagnostics** from the main **Ruleset Editor** (for enabling or disabling function diagnostics for all functions in the ruleset). Click **HERE (Section 3.3.3)** for more information. Examples: • **GetMaxReleaseGivenInflow (% "WattsBar", 0.634 ["1000 * cms"], @"Current Timestep")** • **"GetMaxReleaseGivenInflow" evaluated to: 1299.413 ["cms"]** • **"GetMaxReleaseGivenInflow" constrained to: 1200.00 ["cms"]**
	Init. Rules Print Statements	User defined messages for each initialization rule. Example: • **Entering rule curve routine for January - July**
Data Management	Rule DMI	Execution of rule DMI's for each specified rule.

3.2 Selecting Rules

In the **Rulebased Simulation Diagnostics Configuration**, you are able to select rules for which you wish to see diagnostics. This is done by clicking the **Select** button from the **Show Rules Diagnostics for:** section. If you have both a RBS ruleset and a non-empty Initialization ruleset, you must choose from which set's rules you would like to choose. The Rule Selector then opens. For more information on using this dialog, click **HERE (RPLUserInterface.pdf, Section 6)**.

Also, all rules can be added from the **Rulebased Simulation Diagnostics Configuration**, **Select All ➡ Rules** menu. Clicking **OK** confirms and closes the dialog or clicking **Cancel**, cancel the selection.

3.3 Posting Messages to Diagnostics

There are three ways to generate diagnostics from within RPL: Print Statements, Warning Statements, and Function Diagnostics. All three generate diagnostics messages in the **Diagnostics Output** window. Print Statements and Warning Statements may be added to any rule as a top level statement. They may not be embedded inside of a slot assignment's logic or calculations. Function Diagnostics may be

enabled in predefined and user-defined functions as a before-execution and/or after-execution diagnostic. Function Diagnostics are not user-configurable aside from the selection of units in which values are reported.

3.3.1 Print Statements

The unspecified <expr> of the Print statement can be any concatenation of the different RPL expression types. All of the elements in the Print Statement are interpreted as strings and formatted into the message. In order for the message to appear in the **Diagnostics Output** window, diagnostics and the **Print Statements** diagnostics group must be enabled. They are shown in the diagnostics in a blue text color.

3.3.2 Warning Statements

The unspecified <expr> of the Warning statement can be any concatenation of the different RPL expression types. All of the elements in the Warning Statement are interpreted as strings and formatted into the message. They are shown in the diagnostics in the brown warning text color. Unlike Print Statements, Warning Statements are shown regardless of diagnostics settings.

3.3.3 Function Diagnostics

To understand more about a rulebased simulation, it is sometimes useful to know more about function execution. RiverWare provides two types of diagnostic messages related to function execution:

- Before Execution diagnostic: The diagnostic is issued just before a function is executed and gives the name of the function and the values passed in as arguments.
- After Execution diagnostic: This diagnostic is issued after the function has completed execution and gives the name of the function and the value to which it evaluated.

In order to receive diagnostics for a function, you must do the following:

- Open the Diagnostic Manager (accessible, for example, through the Run Control View menu), enable informational diagnostics and select whether they should go to the Output Window or a file. Note that this step is required in order to receive any informational diagnostics.
- Open the **Rulebased Simulation Diagnostics Configuration** window and toggle on the Function Execution category. Note that this category can be filtered by timestep and rule, if desired.

- Open the Function Editor for the function for which you would like to see diagnostics. Select **View ➤ Show Diagnostic Settings** or click on the **Diagnostic Settings** toggle. This will add a panel to the Function Editor which allows you to toggle on or off either type of diagnostic (**Before Execution** or **After Execution**) for that particular function (see graphic). If the return type of the function is **NUMERIC** and you would like to see **After Execution** diagnostics
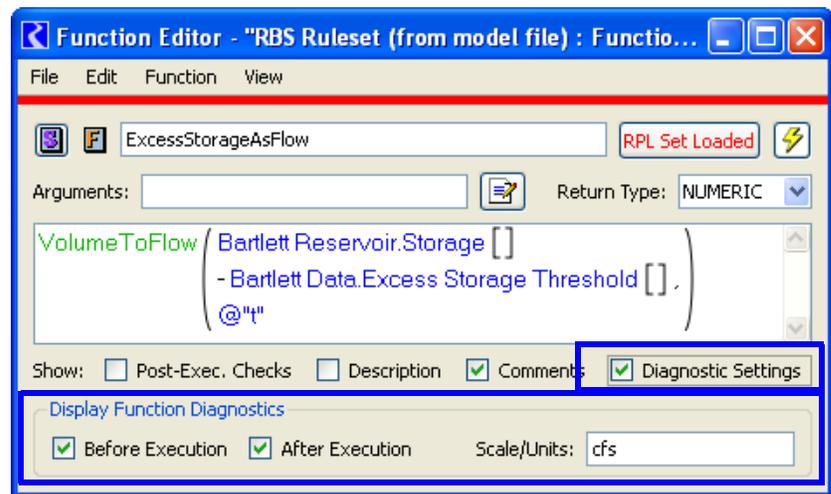
for that function, then you may enter the scale and units in which you would like the function's return value to appear within the diagnostic. Examples include 1000 cfs and ft. No quotes are required.

To enable or disable function diagnostics for many functions at once, from the RPL set editor select **Ruleset ➤ Function Diagnostics**. This will display a sub-menu which allows the following options:

- Enable Before Execution diagnostics for all user defined functions
- Enable After Execution diagnostics for all user defined functions
- Enable Before Execution diagnostics for all predefined functions
- Enable After Execution diagnostics for all predefined functions
- Disable Before Execution diagnostics for all user defined functions
- Disable After Execution diagnostics for all user defined functions
- Disable Before Execution diagnostics for all predefined functions
- Disable After Execution diagnostics for all predefined functions

Optimization Diagnostics

4.   Optimization Diagnostics

The **Optimization Diagnostics Configuration** dialog allows you to select from thirteen diagnostics groups and three context filters which are relevant during a Optimization run. This dialog's settings are only valid when the Optimization controller is selected. It is accessed from the Optimization button on the Diagnostics Manager.
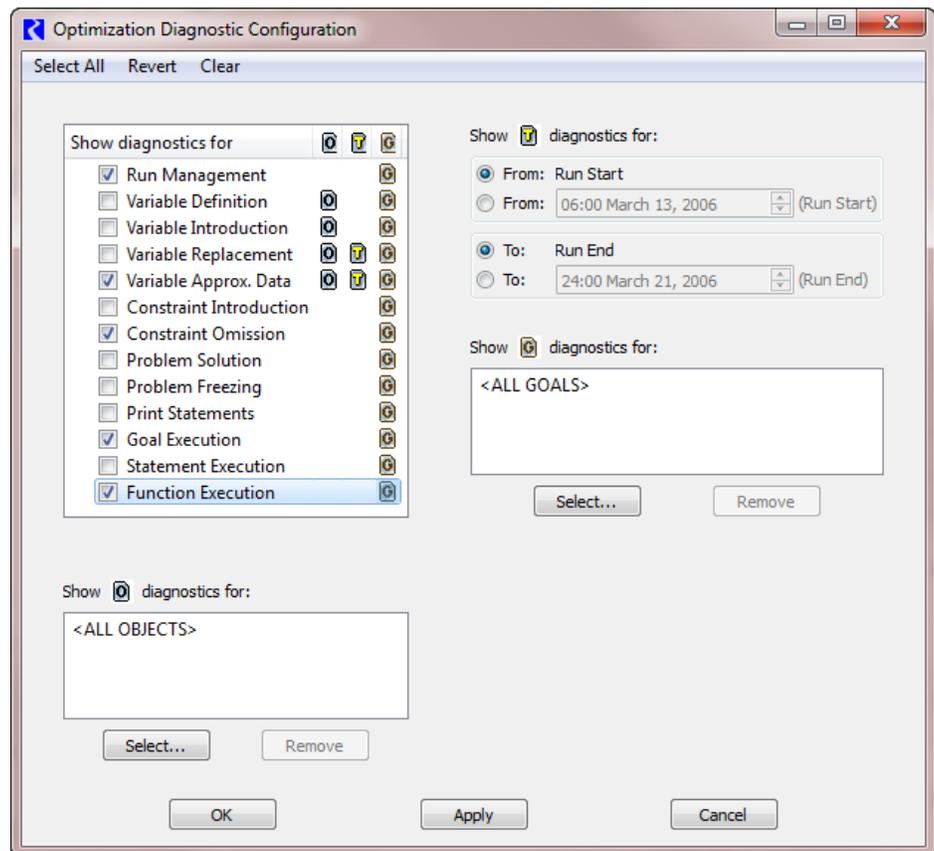
4.1 Optimization Diagnostic Groups

The table below indicates the kinds of messages which are displayed when a Diagnostic Group is selected. Examples are shown in Bold font.

Diagnostics Group	Type of diagnostics issued
Run Management	Beginning/End of Run and each goal. Examples: • **Initializing model run.** • **Beginning of run.** • **Begin goal.** • **Execute goal.** • **End goal** • **End of run.**
Variable Definition	Diagnostic messages for definition of each variable: • **Defining the variables of slot "SM3 reservoir.Pool Elevation" in terms of variables on the shifted slot "Shifted Pool Elevation" (shift = -320.00000000 "m^1").** • **Adding definition "Shift Scaling of Pool Elevation" for the following variables: SM3 reservoir.Pool Elevation [t]** • **... Definition: (SM3 reservoir.Pool Elevation [t] == (SM3 reservoir.Shifted Pool Elevation [t] - -320.00000000 "m^1"))**
Variable Introduction	Diagnostic messages for introduction of new decision variables: • **While preparing for the run, encountered the new decision variable "SM3 reservoir.Outflow [24:00 October 31, 2117]"; to define this variable 1 constraints will be introduced into the optimization problem.** • **While executing statement 3.1.1(date=24:00 October 31, 2092).1, encountered the new decision variable "SM3 reservoir.Opt piece for Power.4 [24:00 October 31, 2092]"; to define this variable 1 constraints will be introduced into the optimization problem.**
Variable Replacement	Diagnostic messages for replacement of variables: • **While executing statement 1.1.1(date=24:00 March 31, 2100).1, replacing one constraint with another.** • **... Constraint being replaced: ((1.00000000 * SM3 reservoir.Shifted Pool Elevation [24:00 March 31, 2100]) <= 87.00000000 "m^1")** • **... Replacement constraint: (SM3 reservoir.Storage [24:00 March 31, 2100] <= 12642800000.00000000 "m^3")**

Diagnostics Group	Type of diagnostics issued
Variable Approx. Data	Diagnostic messages for approximation of data: • **The constraint "((1.00000000 * Wilbur.Bypass Capacity [18:00 March 13, 2006]) >= 1.13267386 "m^3 / s^1")" was determined to be ineffective while attempting to substitute an equivalent constraint on "Wilbur.Storage" using the table "Wilbur.Bypass Capacity Table". The constraint has been ignored because it is redundant for all values in the table, which may need additional values. The nearest value in the table is 1.982179 cms. If this constraint was indirectly generated by RiverWare, prior diagnostic messages or turning on informational diagnostics may be helpful. This warning is issued for this constraint once per run, although this situation might hold for multiple dates. The data table is automatically generated at the beginning of each run based on "Wilbur.Bypass Table", so data corrections should be made there.**
Constraint Introduction	Diagnostic messages for introduction and linearization of constraints: • **Adding the defining constraint "(Def.) Economic Value.Linear Avoided Operating Cost [24:00 September 30, 2085], Linear Avoided Operating Cost Replacement" to the optimization problem.** • **... Original constraint: (SM3 reservoir.Power [24:00 June 30, 2110] >= 100.00000000 "MW")** • **... Linearized constraint: ((((((((3.00965758 "1000000.000000 kg^1 / m^1-s^2" * SM3 reservoir.Opt piece for Power [24:00 June 30, 2110]) + (2.98868019 "1000000.000000 kg^1 / m^1-s^2" * SM3 reservoir.Opt piece for Power.1 [24:00 June 30, 2110])) + (2.97411902 "1000000.000000 kg^1 / m^1-s^2" * SM3 reservoir.Opt piece for Power.2 [24:00 June 30, 2110])) + (2.95792886 "1000000.000000 kg^1 / m^1-s^2" * SM3 reservoir.Opt piece for Power.3 [24:00 June 30, 2110])) + (2.90415559 "1000000.000000 kg^1 / m^1-s^2" * SM3 reservoir.Opt piece for Power.4 [24:00 June 30, 2110])) + (2.77190824 "1000000.000000 kg^1 / m^1-s^2" * SM3 reservoir.Opt piece for Power.5 [24:00 June 30, 2110])) + (2.20884309 "1000000.000000 kg^1 / m^1-s^2" * SM3 reservoir.Opt piece for Power.6 [24:00 June 30, 2110])) >= 100.00000000 "1000000.000000 m^2-kg^1 / s^3")**

Diagnostics Group	Type of diagnostics issued
Constraint Omission	Diagnostic messages when constrains are skipped as they will have no effect. • **Attempt to add a soft constraint "C22.1.1(res=Wilson).1(date=12:00 March 21, 2006).1 70" which would not constrain the problem (e.g., it might be weaker than the restrictions imposed by the bounds on the variables), so it is not being added. The full constraint is: ((1.00000000 * Wilson.Outflow [12:00 March 21, 2006]) >= 0.14158423 "1000.000000 m^3 / s^1")** • **Skipping the new constraint C25.1.1(res=Apalachia).1(date=24:00 March 21, 2006).1 36 because the left-hand side is already bounded by other constraints (C11.1.1(res=Apalachia).1(date=24:00 March 21, 2006).1 1 A and C11.1.1(res=Apalachia).1(date=24:00 March 21, 2006).1 1 B). Currently, 67204983.66069999 "m^3" <= lefthand side <= 67204983.66069999 "m^3".**
Problem Solution	Diagnostic messages for solving the optimization problem and the number of variables and constraints that are freezing • **Freezing 0 variables.** • **Freezing 0 constraints.** • **While executing statement 2.1, defining a REPEATED MAXIMIN objective involving 2 constraints and solving the problem.** • **... Objective: (100.00000000 * ZM2.1)** • **... Original objective: (...** • **... Linearized objective (...**
Problem Freezing	Diagnostic messages for freezing variables and constraints • **Frozen constraint: Ending Pool Elevation, C2.1.1(date=24:00 September 30, 2119).1 1 B: was 267.970000 <= ... <= 267.970000 (dual cost = 30.567953). Current statement: CONSTRAINT $ "SM3 reservoir.Pool Elevation" [date] == $ "Opt Data.Ending Elevation" [date]** • **Frozen variable: ZM3.1 = 0.100000 (reduced cost = 100.000000)**
Print Statements	User defined messages within rules for each goal. Example: • **Minimum Load goal was successful.**
Goal Execution	Diagnostic messages displaying when a goal is executed: • **Executing goal #3 ("MinimumLoad", within group "Firm Energy")**

Diagnostics Group	Type of diagnostics issued
Statement Execution	Diagnostic messages displaying when each statement within a goal is executed • **Executing CONSTRAINT statement 3.1.1(date=24:00 September 30, 2119).1, which will attempt to add the soft constraint: (SM3 reservoir.Power [24:00 September 30, 2119] >= 100.00000000 "MW").** • **Executing REPEATED MAXIMIN statement 1.1.**
Function Execution	Diagnostics messages for function execution (both user defined and pre-defined). "Before Execution" function diagnostics show the function being called and the arguments passed into the function. "After Execution" function diagnostics show the value that the function evaluated to and whether or not the value was constrained. NOTE: Function diagnostics also need to be enabled on the functions themselves. This can be done through the **Function Editor** dialog (for enabling diagnostics on individual functions) or by selecting **Ruleset ➥ Ruleset Function Diagnostics** from the main **RPL Set Editor** (for enabling or disabling function diagnostics for all functions in the ruleset). Click **HERE (Section 3.3.3)** for more information. • **GetMaxReleaseGivenInflow (% "WattsBar", 0.634 ["1000 * cms"], @"Start Timestep")** • **"GetMaxReleaseGivenInflow" evaluated to: 1299.413 ["cms"]**

Workspace

5.   Workspace

The Workspace diagnostic settings contain two **Data Management** groups; **File** and **Slot** (with Slot 📄 context). The **File** group includes diagnostics which are posted by a DMI; the **Slot** group includes diagnostics which are posted by a slot when it's importing or exporting a data file.

Note: This document is under development.

5.1 Workspace Diagnostic Groups

The table below indicates the kinds of messages which are displayed when a Diagnostic Group is selected. Examples are shown in Bold font.

Super Group	Diagnostics Group	Type of diagnostics issued
Model Load/ Save:		
Data Management		
	DMI Management	
	DMI Invoke	
	DMI Dataset	
	DMI Slot	
	Rule DMI	
Workspace Management:		
Controller Management:		Any changes to controller settings: • **Controller set to "Rulebased Simulation".**
Rule Management Groups:		
Subbasin Management		Any changes in Subbasin management • **Add subBasin ("New Subbasin 1").**

| Super Group	Diagnostics Group	Type of diagnostics issued
Object Management		Any changes to an Object on the workspace • **OBJECT: MuddyLake: Rename object to "MuddyReservoir".**
Slot Management		
Link Management		
Dependency Management		
Name Map Management		
Client Server		
MRM Rule Execution		For the iterative MRM ruleset, this prints information on MRM rule, timestep, and slot. This is similar information to Rule Execution **HERE (Rule Execution)**.
MRM RPL Function Execution		For the iterative MRM ruleset, prints before and after execution diagnostics for user and pre-defined functions. This provides similar information to Function Execution diagnostics **HERE (Function Execution)**.
MRM RPL Print Statements		For the iterative MRM ruleset, this allows print statements to post diagnostics. This is similar information to Print Statement **HERE (Print Statements)**.
MRM Rule DMI		For the iterative MRM ruleset, this posts information on DMI's that are executed by MRM Rules.

Super Group	Diagnostics Group	Type of diagnostics issued
	Expr. Slot Execution	Expression slot execution diagnostics are printed for expression slots that are evaluated outside of a run using the **Evaluation ➥ Evaluate** menu on the slot or the **Control ➥ Evaluate Expression Slots** menu from the workspace. Note, there is no timestep filtering on the workspace diagnostics, so these diagnostics will print a message for all timesteps in which the expression slot evaluates. Example: • **Expression evaluated to 624.00 [1000 acre-ft]**
	Expr. Slot Function Execution	Evaluation of RPL functions called by expression slots evaluated outside of a run. Diagnostics messages are printed for function execution (both user defined and pre-defined). "Before Execution" function diagnostics show the function being called and the arguments passed into the function. "After Execution" function diagnostics show the value that the function evaluated to and whether or not the value was constrained. NOTE: Function diagnostics also need to be enabled on the functions themselves. This can be done through the **Function Editor** dialog (for enabling diagnostics on individual functions) or by selecting **Set ➥ Function Diagnostics** from the main **Set Editor** (for enabling or disabling function diagnostics for all functions in the set). The Expression Slot Set editor is accessed from the Workspace **Policy ➥ Open Expression Slot RPL Set**. Click **HERE (Section 3.3.3)** for more information on configuring the diagnostics or **HERE (Slots.pdf, Section 4.6)** for more information on expression slots. Example: • **SumFlowsToVolume($"Mountain Storage.Inflow", @"Start Timestep", @"t")** • **"SumFlowsToVolume" evaluated to: 274839.74 ["m3"].**

6.   Accounting Diagnostics

To the right is a sample of the **Post-Simulation Accounting Diagnostics Configuration** dialog.

Available filters include Objects, Accounts, Slots, and Timesteps. Diagnostics allow the user to see when each individual account is solving, what slots are known and unknown at that timestep, the accounting method being used, what the value is when an account's slot is set, and more.
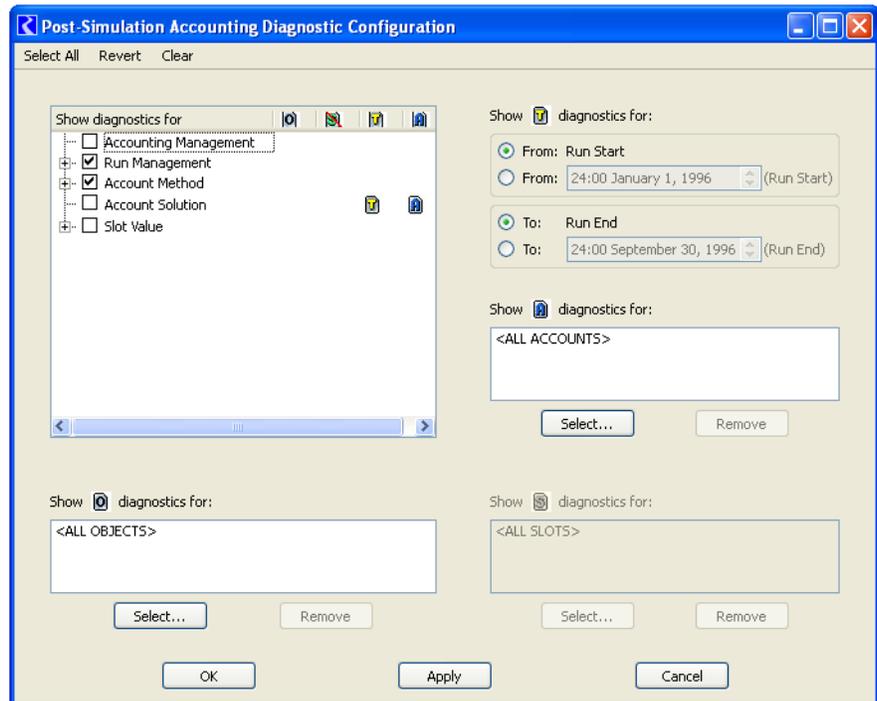
6.1 Accounting Diagnostic Groups

The table below indicates the kinds of messages which are displayed when a Diagnostic Group is selected. Examples are shown in Helvetica Bold font.

Super Group	Diagnostics Group	Type of diagnostics issued
Run Management:		
	Controller	Beginning/End of Run/Timestep for each time. Examples: • **Initializing model run.** • **Beginning of Run.** • **Begin Timestep.** • **Execute timestep.** • **End of run.**

Super Group	Diagnostics Group	Type of diagnostics issued
	SimObj	Beginning/End of Run/Timestep for each Object and time. Examples: • **Clear state.** • **Propagating user input.** • **Beginning of run.** • **Begin timestep.** • **End timestep.** • **End of run.**
	Account	Beginning of Run for each Account and time. Examples: • **Clear state.** • **Propagating user input.**
Account Method:		
	Controller	Describes when the controller is notified of a change to an account's slot. Example: • **The "Post-Simulation Accounting" controller received a slot changed notification for slot "BigLake^City.Inflow".**
	SimObj	Describes which user defined methods are being executed. Example: • **Executing Category "Storage Account Gain Loss" with selected method "BigLake_Gain Loss".**
	User Method Execution	Diagnostics for object level accounting methods. Examples: • **Executing user-defined accounting method #3 ("HappyLake_SlotInflows", within group "Storage Account Slot Inflow")** • **Assignment initiated (the left-hand side is "$ "HappyLake^Fish.Slot Inflow" []").** • **Evaluation of Assign statement successful; will attempt assignment: Fish.Slot Inflow[January 1, 2004] = 3.99 [1.00 * cfs].**
Independent Groups:		
	Account Management	• N/A

Super Group	Diagnostics Group	Type of diagnostics issued
	Account Solution	Describes when the account is solving including the knowns and unknowns. Example: • **Inflow is known, Storage is allowed, Outflow is input; solving for Storage.**
Slot Value:		
	Set Value	Describes when an accounting slot is set.. Example: • **Confluence^Fish.Inflow: Set value = 25.740254 * 1.0cfs**
	Interpolation	TableSlot table interpolation lookups for each account slot.