Optimization Software Enhancements

Summer 2012

Authors: Patrick Lynn, Phil Weinstein

This document describes RiverWare enhancements designed to improve usability of the optimization system as well as the maintainability of the software supporting that system. These improvements were completed in Summer 2012 for the TVA (Tennessee Valley Authority) and will be included in RiverWare version 6.3.

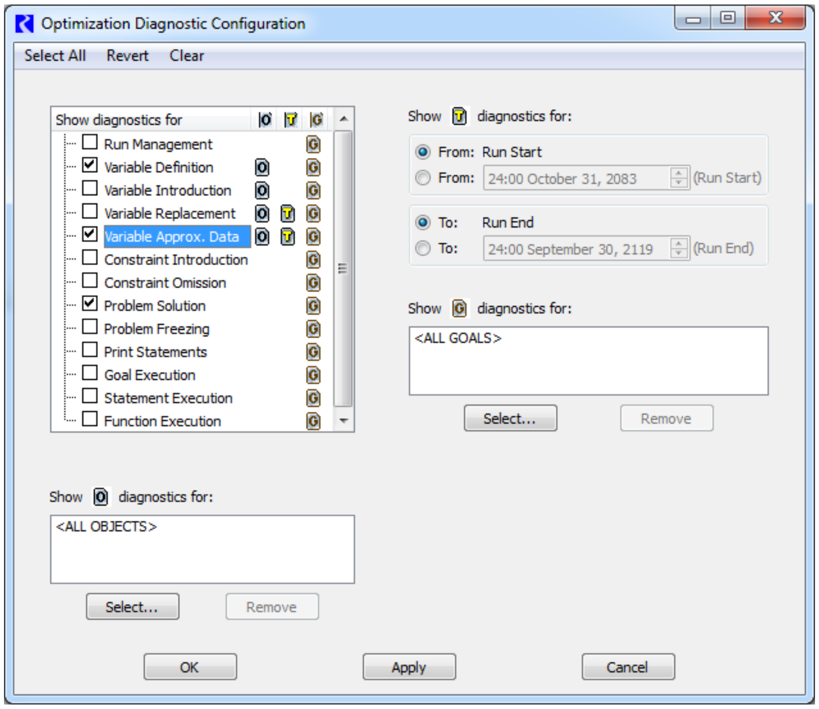
1 Diagnostics

1.1 The Optimization Diagnostic Configuration dialog

Since RiverWare's optimization controller was redesigned, it has provided only limited support for diagnostics. The Optimization Diagnostic Configuration dialog was reinstated, existing messages were improved, new messages were created, and all optimization messages were organized into the following new categories:

- Run Management
- Variable Definition
- Variable Introduction
- Variable Replacement
- Variable Approximation Data
- Constraint Introduction
- Constraint Omission
- Problem Solution
- Problem Freezing
- Print Statements
- Goal Execution
- Statement Execution
- Function Execution

Figure 1. The new Optimization Diagnostic Configuration dialog.



Show diagnostics for

- [ ] Run Management
- [✓] Variable Definition
- [ ] Variable Introduction
- [ ] Variable Replacement
- [✓] Variable Approx. Data
- [ ] Constraint Introduction
- [ ] Constraint Omission
- [✓] Problem Solution
- [ ] Problem Freezing
- [ ] Print Statements
- [ ] Goal Execution
- [ ] Statement Execution
- [ ] Function Execution

Show diagnostics for:
[✓ From: Run Start]
From: 24:00 October 31, 2083 (Run Start)
[✓ To: Run End]
To: 24:00 September 30, 2119 (Run End)

Show diagnostics for:
<ALL GOALS>

Show diagnostics for:
<ALL OBJECTS>

The diagnostic categories allow users to selectively filter the large volume of diagnostic messages down to the messages of interest. The users can filter appropriate categories by an object, time period, and/or goal of interest.

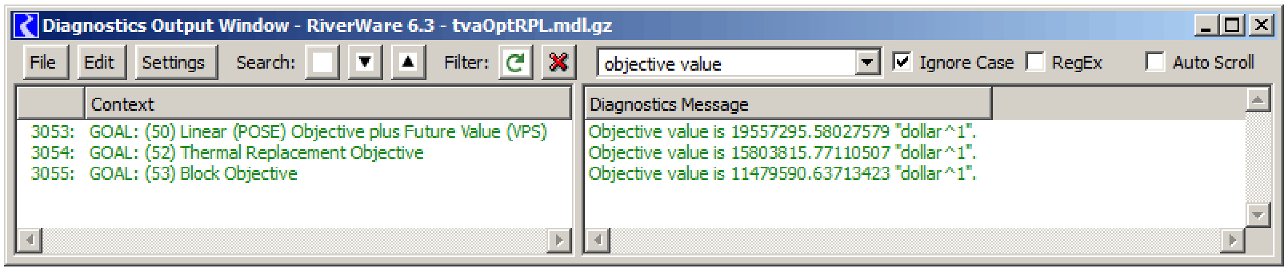
As part of this work, the RPL set editor was modified to support the inclusion of Print statements within goals.

1.2 Diagnostic Output dialog message filtering

The RiverWare Diagnostics Output Window has been enhanced to provide user-controlled *display filtering* of diagnostics messages. Filtering is supported in parallel with this dialog's previously available "search" capabilities. For filtering, only *character string* matching is supported -- that is, not the search feature's *message type* matching (e.g. warning vs. error) capability.

Additional enhancements to the Diagnostic Output Window include:

- Improvements in pattern matching capabilities for both search and filter functions. This includes support for non-case-sensitive matches, "regular expression" pattern syntax, and improved support for "wildcard" pattern syntax.
- Search pattern history menu (implemented using an editable combo box).
- Automatic width adjustment options.



Context	Diagnostics Message
3053: GOAL: (50) Linear (POSE) Objective plus Future Value (VPS)	Objective value is 19557295.58027579 "dollar^1".
3054: GOAL: (52) Thermal Replacement Objective	Objective value is 15803815.77110507 "dollar^1".
3055: GOAL: (53) Block Objective	Objective value is 11479590.63713423 "dollar^1".

The sort of filtering supported with this enhancement -- *display* filtering -- has a temporary effect on the display of the internally maintained set of generated diagnostic messages. It does not remove any generated diagnostic messages, nor does it prevent the generation of any particular new diagnostic messages. All generated messages can be displayed again by clearing the filter, and a different filter operation can be applied (in place of a previously applied filter).

When applying a display filter to messages, all previously generated messages "failing" (not matching) the specified filter pattern are hidden, as are newly generated messages failing the pattern. A diagnostic message matches the pattern if the "context" and "message" components of the message -- together -- *contain* the matching pattern.

The following "high priority" messages are always shown, regardless of the active filter specification:

- Errors
- Internal Errors

These enhancements to the Diagnostics Output dialog are described in more detail in this document:

Diagnostics Output Message Display Filtering / Version 1 for RiverWare 6.3
8-31-2012, 6 pp. [source dir: R:\doc\diagnostics\DisplayFiltering\]

1.3 Optimization controller warnings

Several improvements were made that involved warnings and confirmation dialogs:

- The confirmation dialog presented to the user when switching to the Optimization Controller within the Run Control dialog was removed.
- When the user initiates a run from either the Run Control dialog or the SCT, we now try to identify situations in which the user mistakenly omitted a run in the typical optimization run sequence (Sim -> Opt -> RBS). If RiverWare notices an anomaly, the user is presented with a dialog asking them if they want to continue. There are two specific scenarios that we catch:
    - A Rulebased simulation run followed by an Optimization run.
    - A Simulation run followed by a Rulebased Simulation run (when there is an optimization policy set loaded).
- If a slot is expected to be approximated using piecewise approximation and the secant approximation is used instead, RiverWare now issues a warning.
- Several warnings related to approximation data tables were issued many times for some models. These warnings are now issued only once per run for a given approximation and thereafter are issued as a regular (Information) diagnostic.

2 Performance analysis and improvements

CADSWES staff performed a run-time performance analysis and discovered no high-impact inefficiencies. However, memory analyses indicated large memory growth within a run as well as between runs within the same execution of RiverWare. CADSWES staff identified and fixed a few true memory leaks as well as a few instances of inefficient use of CPLEX memory resources, thereby greatly reducing memory growth within most runs. Now, for runs after the first run, memory growth within the run is close to zero.

To illustrate the impact of these changes, consider the figures below, corresponding to batch runs of RiverWare before and after the memory improvements were made. For each RiverWare execution, a small (one reservoir) model was loaded and run in the Optimization controller several times. For this model's optimization policy, the bulk of the run time (~27 s) is spent executing a Repeated Minimax that iterates 424 times. In each figure, a screen snapshot of a monitoring tool shows the growth in memory of the RiverWare process, with spikes in I/O indicating the end of each run.

Figure 1. Batch execution of RiverWare, before improvements.

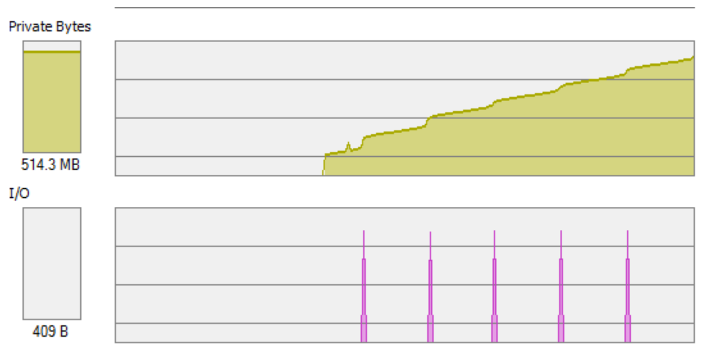
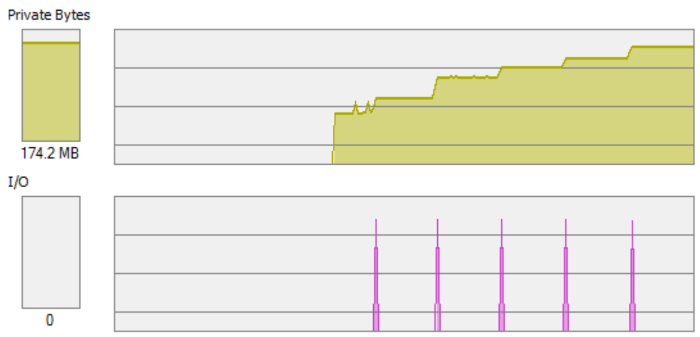


Figure 2. Batch execution of RiverWare, after improvements.



Note that before the memory improvements were applied, memory usage exhibited a jump at the beginning of each run, followed by even greater magnitude growth within each run, and the within-run growth is almost completely eliminated by the fixes. For this model, the final batch run memory usage has decreased from 514 MB to 174 MB, a 2/3 reduction in virtual memory.

3 AggSeriesSlot Conversion

Prior to the redesign of the optimization system, slots referenced by Optimization policies typically were required to be AggSeriesSlots, where the second and third columns were reserved for optimization input and output. The redesign of the optimization system relaxed this requirement, only the first column of these slots is now used by optimization. To simplify both the user interface displaying these slots and the software accessing them, these slots were converted to SeriesSlots. Of the 191 AggSeriesSlots used within RiverWare, 95 of them have now become SeriesSlots. When loading an older model with the next release, users will be notified through diagnostics of the conversion process and that these slots will subsequently be saved as SeriesSlots.

4 RPL

The editing of RPL policy was improved in the following ways :

- RPL Palette button that do not apply in the context of an optimization goal set (e.g., the Flag Value buttons) are now disabled. In addition, RiverWare now conducts a more thorough analysis when an expression is selected within a goal, allowing additional buttons to be disabled when they do not apply to in the specific context of the selection. For example, the top-level expression in an Add Constraint statement is restricted to the following operators: >=, <=, and ==. Eliminating palette choices that don't make sense makes it easier for users to build the policy they have in mind.
- The specification of a slot name for the right-hand side of an object/slot expression was streamlined by enabling the RPL palette’s slot selector button in this context. When the user makes a slot selection, its name is used to fill in the object/slot expression’s right-hand side.

5 Future work

The task focused on the enhancements that CADSWES thought were most important for TVA. After completing those, we were able to complete some, but not all, of the tasks we had identified as lower priority. In the process of working on this task we identified additional improvements that could be made in the future. These new tasks and the lower priority tasks that were not completed are listed below. These items could be bundled into a task for FY 2013.

- Display typical run sequence and status in that sequence.
- New predefined functions to simplify optimization policy (e.g., DatesInRun()).
- Prevent accidental deletion of a goal (possibly by implementing Undo for set editor).Enable object & slot selectors for creation of a list expression.
- Enable arrow keys for rearranging items within list expressions.
- Deal gracefully with unintended inline editing within expressions.
- Issue warnings/errors when Freeze statement is missing or occurs at incorrect location.
- Write an OptExpr method which checks for unit conformance in an OptExpr (Values have RplUnits for values andwe can get them for variables).
- Add scale to Opt::num().
- Finish support for definitions which contain numeric values with time-varying units.
- Improve GUI for SUM/AVE.
- Deal with shrinking a hard constraint to a soft and vice versa.
- When we add a non-decision variable to the problem, if it has bounds, add appropriate constraints (these can be considered to be additional defining constraints).
- Deal more comprehensively with illegal CPLEX characters in names.
- Test and support CPLEX use of multiple cores.