

Technical Documentation Version 8.0

Data Management Interface (DMI)



Center for Advanced Decision Support for Water and Environmental Systems (CADSWES)

UNIVERSITY OF COLORADO **BOULDER**

These documents are copyrighted by the Regents of the University of Colorado. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means electronic, mechanical, recording or otherwise without the prior written consent of The University of Colorado. All rights are reserved by The University of Colorado.

The University of Colorado makes no warranty of any kind with respect to the completeness or accuracy of this document. The University of Colorado may make improvements and/or changes in the product(s) and/or programs described within this document at any time and without notice.

Contents

1. General DMI Information 1
   - Types of DMIs 1
   - DMI User Interface 2
     - Accessing DMIs and the DMI Manager 2
     - DMI Manager Dialog 2
       - Copy and Paste DMIs 5
       - Creating, Deleting, and Sorting DMIs 5
       - DMI Groups 6
       - Year Mapping Groups 7
       - Locking the DMI Manager Dialog 11
     - DMI Parameter Dialog 11
       - Creating and Deleting Parameters 12
       - Editing a Parameter 13
       - Editing a Parameter Name 13
       - Editing a Parameter Type and Attributes 13
       - Exporting and Importing Parameters 16
       - Reconciling Parameter Edits 17
     - DMI Invocation Manager Dialog 18
       - Overview 19
       - Views 20
       - Clearing Values 21
       - Deleting Invocation Records 22
     - Invoking DMIs and DMI Groups 22
       - Warning and Error Messages 24
       - DMI Status and Progress 24
     - Icon Quick Reference 27

2. Control File-executable DMI 29
   - About the Control File-executable DMI 29
   - Control File 30
     - Object.Slot Specifications 31
     - File Name and Directory Specification 33
     - Optional keyword=value Pairs 34
     - User keyword=value Pairs 36
   - Data Files 37
   - DMI Executable 40
   - DMI Editor Dialog 42
     - Name 42
     - Type 43

Confirm Warnings Checkbox 43
Record Invocations 43
Allow Spaces in File Paths 44
Limit Output to Run Range 44
Line Ending 44
Control File 45
Executable 45
User Parameters 46
Command Line 47

3. Trace Directory DMI 49

About the Trace Directory DMI 49

Trace Directory DMI Editor Dialog 51

4. Database DMI 55

About the Database DMI 55

Creating a Database DMI 58

Name and Type 61
DMI Configuration 62
Dataset 63
Slot Selection 64
Slots in Selection 66
Enabling and Disabling for Debugging 68
Use of Name Maps 68
Unused Slot Selections 69
DSS Dataset Configuration 69
Setting Multiple DSS Dataset Files 70
DSS DMI User Parameters 71
Examples 72
Example 1: DSS DMI Using the Database DMI Editor 72
Example 2: Excel DMI to Export Data Using Headers 74
Example 3: Excel DMI to Import Data Using Ranges 77

Datasets Overview 82

DSS Datasets 83

DSS File 83
Name Map 83
Missing Values 84
Table Slot Data 84
Units 84
Data Types 85

HDB Datasets 86

General Tab 86
Name Map 86
Missing Values 86
Group Slots for Writing 87
Units 87

HDB Tab 88
HDB Database 88
Mapping ID 89
HDB Table Type 90
HDB Table Type—Model 90
HDB Table Type—Real 93
HDB Table Type—Ensembles 96
Excel Datasets 100
General Tab 100
Name Map 100
Missing Values 101
Units 101
Excel Tab 103
Workbook 103
Slot Mapping 104
Slot Mapping—Spreadsheet Ranges Approach 105
Slot Mapping—Header Text and Sheet Names Approach 109
Connection Process 116
Name Mapping 117
Configuring Name Mapping 117
Name Map Ordering and Priorities 119
Sharing Database DMIs 120
Export 120
Import 121

Contents

Chapter 1
General DMI Information

The RiverWare Data Management Interface (DMI) provides a mechanism to transfer data into and out of RiverWare in a highly configurable and automated manner. For example, you can configure a DMI that, with the click of a button, brings all of the observed inflows and initial pool elevations into the model.

Topics

- Types of DMIs, page 1
- DMI User Interface, page 2

Types of DMIs

There are three types of DMIs.

Control File-Executable

In the Control File-Executable approach, the user develops a "control file" that specifies the data to move into or out of RiverWare. Also, the user develops an external "executable" which interacts with the control file, RiverWare, and the external data repository. See "Control File-executable DMI" on page 29.

Trace Directory DMI

The Trace Directory DMI is similar to the Control File-Executable approach where the user develops a "control file" that specifies the data to move into or out of RiverWare. However, instead of an executable, the user specifies a top-level directory, under which the data files for traces are held in subdirectories named with the trace number. See "Trace Directory DMI" on page 49.

Database DMI

Currently, RiverWare supports a direct connection between three databases: the Hydrologic Engineering Center's Data Storage System (HEC-DSS), the Hydrologic Data Base (HDB), and Microsoft Office Excel. In this configuration, all of the slot specification is done within RiverWare using Datasets, Name Maps, and Database DMIs. See "Database DMI" on page 55.

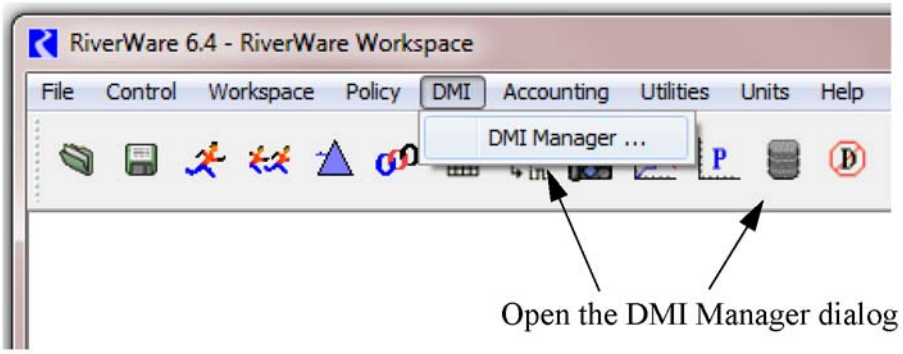
In this section, we describe the DMI user interface components that are general to all of the approaches. Then we describe the Control File-Executable approach and its specific user interface components and the Trace Directory DMI approach and its specific user interface components. Finally, we describe the Database DMI approach and its specific user interface components.

DMI User Interface

This section describes the user interface used to execute and create DMIs. This section applies to all of the DMI approaches. See "Database DMI" on page 55 for details on the user interface for the Database DMI.

Accessing DMIs and the DMI Manager

The Workspace provides a DMI menu and a DMI toolbar button.  Selecting either the **DMI** menu, then **DMI Manager** or the toolbar button opens the DMI Manager dialog. Once one or more DMIs are defined, the DMI menu also contains these DMIs, which can be invoked directly from the workspace. See "Invoking DMIs and DMI Groups" on page 22 for details.

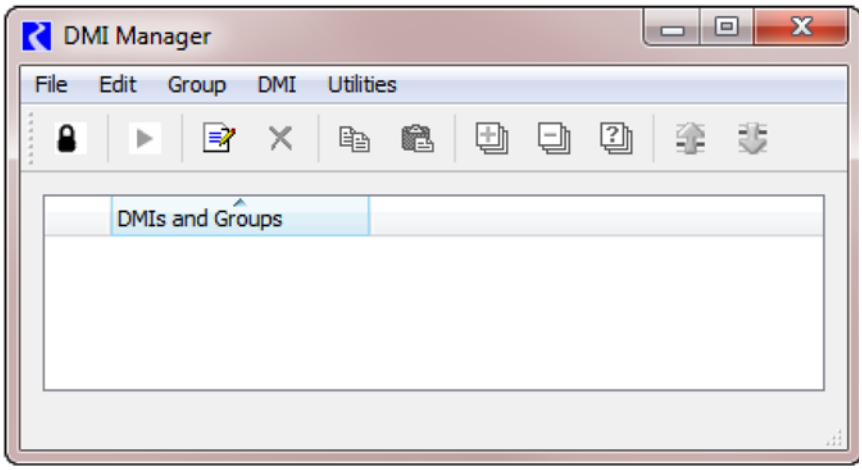


DMI Manager Dialog

The DMI Manager dialog enables you to do the following:

- Create and delete DMIs
- Open the DMI Editor dialog
- Create and delete DMI groups
- Edit DMI groups
- Sort DMIs and DMI groups
- Invoke DMIs and DMI groups
- Copy and paste DMIs and DMI groups
- Open the DMI Parameter dialog
- Lock the DMI Manager

- View records of DMI invocations



In the DMI Manager and related dialogs, actions can be performed using one or more of the following, each of which acts on the current selection:

- A main menu item
- A toolbar button
- A context (right mouse button) menu item
- A keyboard accelerator

Main menu items, toolbar buttons and context menu items performing the same action share the same icon. For example, Figure 1.1 and Figure 1.2 illustrate that from the DMI Manager dialog a user can invoke the selected DMI identically using the **DMI**, then **Invoke** main menu item, the **Invoke** toolbar button, the DMI **Invoke** context menu item and the **Control-I** keyboard accelerator.

Figure 1.1

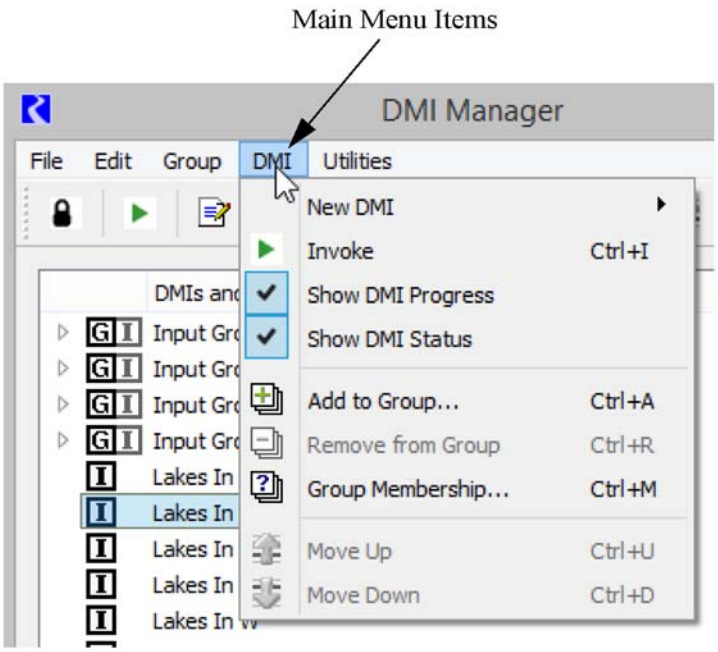
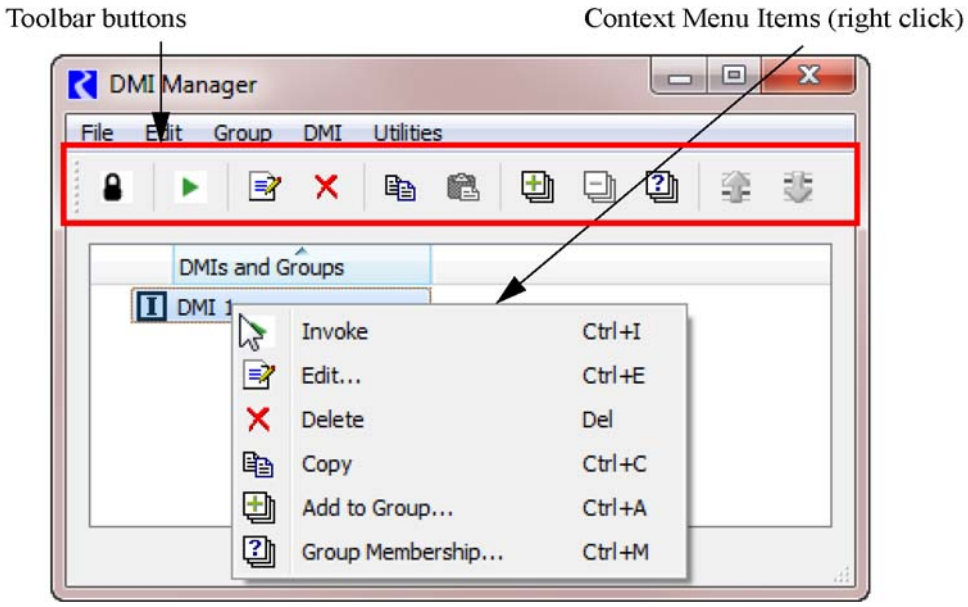


Figure 1.2



Actions which aren't valid for the current selection are disabled. (That is, their main menu items, toolbar buttons and context menu items are disabled.)

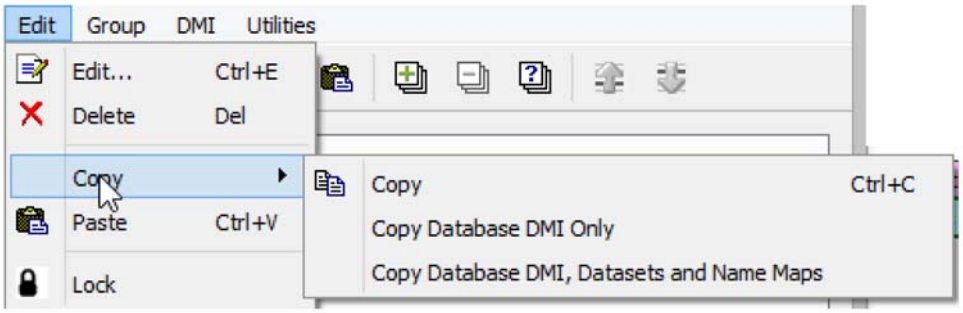
The document refers to main menu items when describing actions; be aware that in most cases toolbar buttons, context menu items and keyboard accelerators are also available for the actions.

Copy and Paste DMIs

The **Edit** menu and right-click context menu have a **Copy** operation. This leads to the following options:

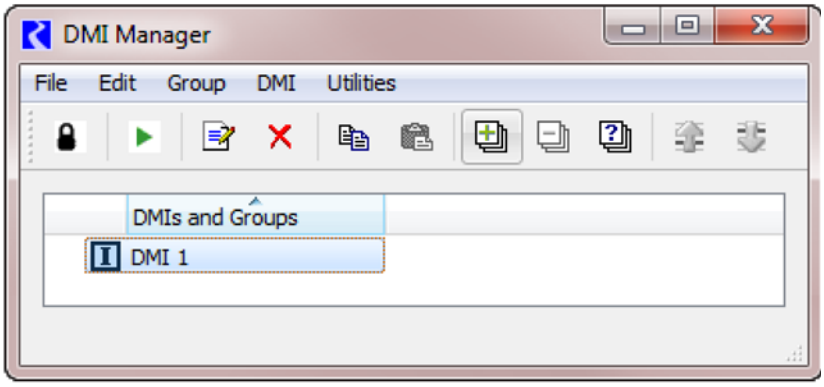
- **Copy**. For Control File - Executable DMIs, Trace Directory DMIs, and DMI groups, this is the standard copy operation. Only the DMI is copied.
- **Copy Database DMI Only**. For Database DMIs, this is a basic copy operation of the Database DMI only. None of the accessed Datasets or Name Maps are copied, only the Database DMI itself. When pasted, the new Database DMI refers to the original Datasets and Name Maps.
- **Copy Database DMI, Datasets, and Name Maps**. For Database DMIs, this operation provides a "deep copy" of the Database DMI and all Datasets and Name Maps accessed. When pasted, the new Database DMI is connected to the newly pasted Datasets and similarly, Name Maps.

A **Paste** operation will add the most recently copied items with the name "Copy of DMI Name 1".



Creating, Deleting, and Sorting DMIs

Create a new DMI under the **DMI**, then **New DMI** main menu item and choose the type of DMI to create; the DMI has a default name.



I indicates an input DMI; similarly O indicates an output DMI.

Delete a DMI with the **Edit**, then **Delete** main menu item.

Open the DMI Editor dialog with the **Edit**, then **Edit** main menu item or double-click the row. The DMI Editor is different for the different types of DMIs. See "DMI Editor Dialog" on page 42 for details on the DMI Editor dialog for the Control File-Executable approach. See "Trace Directory DMI Editor Dialog" on page 51 for details on the

Trace Directory DMI. See “Creating a Database DMI” on page 58 for details on the DMI Editor for the Database DMI.

Sorting of DMIs and DMI groups is limited to sorting by name or type by selecting column header.

DMI Groups

A DMI group is an ordered collection of one or more DMIs. When a DMI group is invoked, its DMIs are invoked in order, until all have completed or one has failed. There are two kinds of groups:

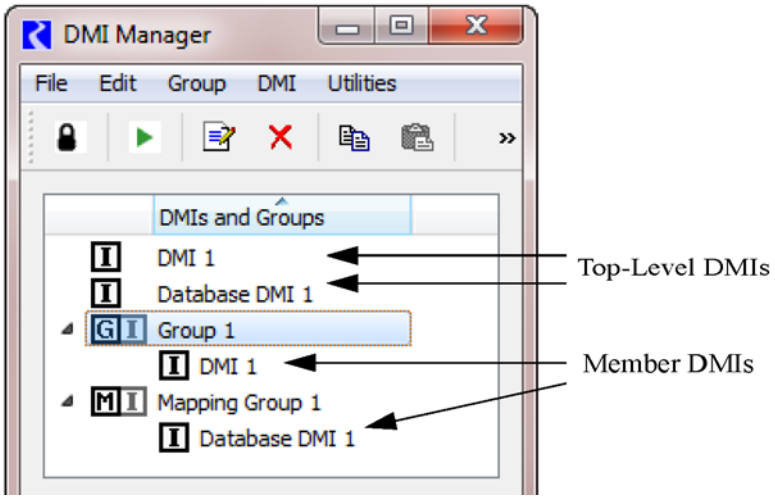
- **DMI Group.** This is a collection of DMIs to be executed in order.
- **Year Mapping Group**. This special type of group maps run years to years in the database. See “Year Mapping Groups” on page 7 for details.

Additionally, an icon indicates whether the group contains all Input DMIs, all Output DMIs or a mix of Input and Output DMIs

Create a new DMI group with the **Group**, then **New Group** or **New Year Mapping Group** main menu item.

DMIs can appear one or more times in the DMI Manager dialog; once as a top-level DMI and zero or more times as a member of a group. Actions which pertain to a DMI globally (for example, editing it) are only valid when a top-level DMI is selected; actions which pertain to a DMI within a group (for example, moving it up or down) are only valid when a member DMI is selected. See Figure 1.3.

Figure 1.3



When the distinction is important, a DMI is referred to as either a top-level DMI or a member DMI.

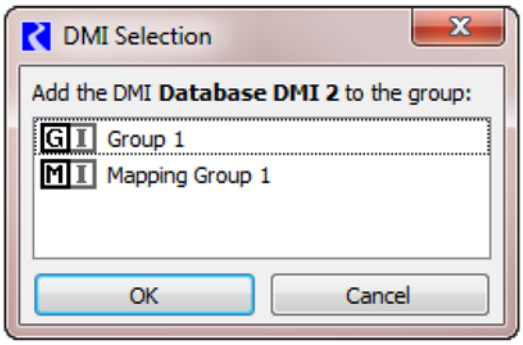
Editing DMI groups includes:

- Editing a DMI group’s name
- Adding a top-level DMI to a group
- Removing a member DMI from a group

- Moving a member DMI within a group (e.g. ordering the DMIs within the group)
- Querying a DMI's group membership

To edit a DMI group's name double-select the name cell, to edit the name in place.

To add a top-level DMI to a group, select the **DMI**, then **Add to Group** main menu item. You are then presented with a dialog listing the available groups; select the desired group and select **OK**.

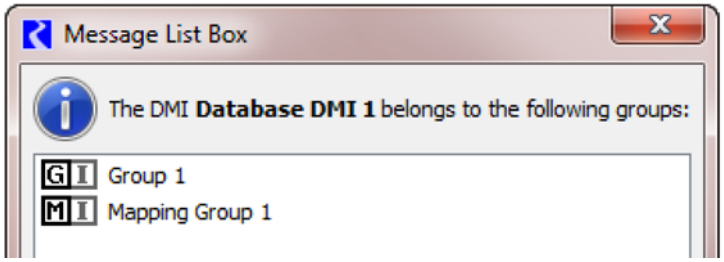


If no groups are available (e.g. there are no groups or the DMI is a member of all groups) you will be notified.

To remove a member DMI from a group, select the **DMI**, then **Remove from Group** main menu item.

To move a member DMI within a group, select either the **DMI**, then **Move Up** main menu item or the **DMI**, then **Move Down** main menu item.

To query a group membership for a DMI, select the **DMI**, then **Group Membership** main menu item; the group membership is displayed.



Year Mapping Groups

A **Year Mapping Group** allows you to import data from the specified years in your database to years in your run. For example, you might want to run your 5 year run (e.g. run dates from 1/1/2000 to 12/31/2005) with data from 1992, 1936, 1952, 1962, 1932. This tool allows you to create the map between the data in the database and the years of your run. It has a specific algorithm for dealing with leap days; see "Year Mapping Algorithm" on page 10 for details.

Before we describe it in detail, following are the limitations on this utility:

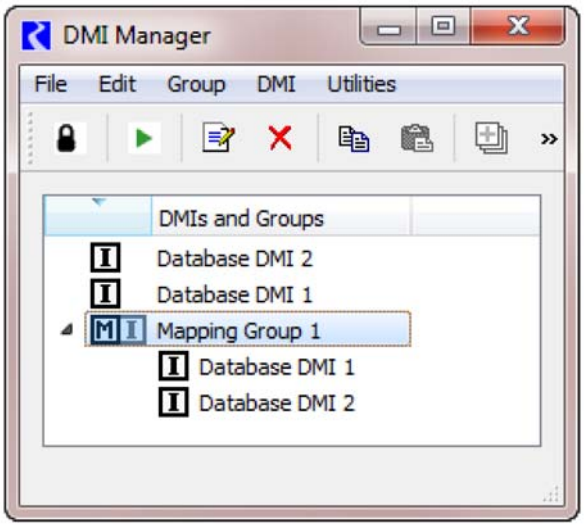
- **Mapping groups are available for 1 Day timestep models only.** Attempting to access mapping groups (creating, editing, invoking) with a model that has a timestep other than 1 Day, results in an error. The model run can start and finish on any day of the year. (Prior to RiverWare 6.9, the run had to start on Jan 1st and end on Dec 31st.)

- **Mapping groups may only contain Input Database DMIs** (DSS, HDB, or Excel)

You create the Year Mapping Group, add Input DMIs to the group, define the mapping, and then invoke the group. These steps are described in the following section:

Create a Year Mapping Group

Year Mapping Groups are added from the DMI Manager's **Group**, then **New Year Mapping Group** menu item.



Adding DMIs to the Year Mapping Group

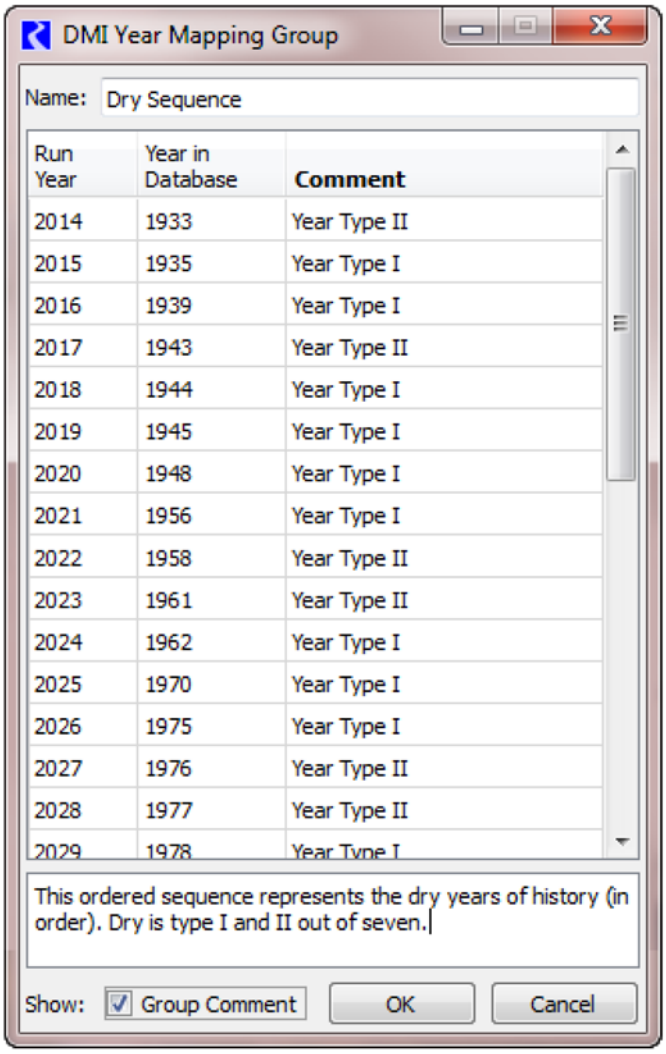
As described above, to add a DMI to a group, select the **DMI**, then **Add to Group** main menu item. You are then presented with a dialog listing the available groups; select the desired group and select **OK**. Remember, only Input Database DMIs may be added to Year Mapping Groups; other types of DMIs will not be shown when you add to the Year Mapping Group.

Defining the Map

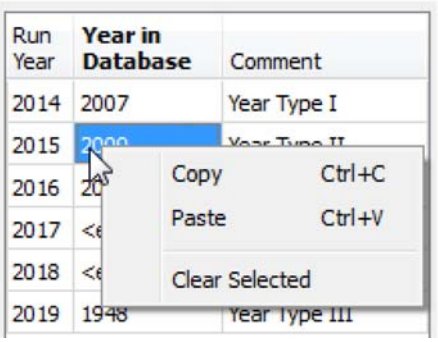
Edit the group by double-clicking the item. This brings up a simple dialog where you may edit the mapping group's name, the **Year in Database**, and **Comment** for each run year.

- **Run Year:** The year in the RiverWare run. This is not editable and is taken from the Run Control.
- **Year in Database:** The year of data you wish to import for the given Run Year.

- **Comment:** A text field to annotate the entry.



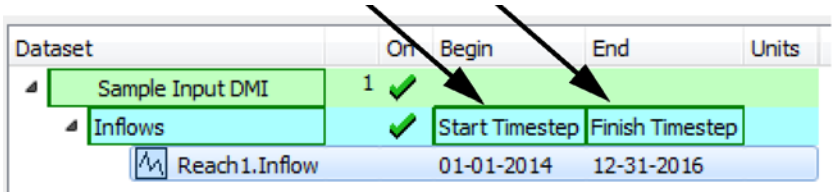
You can copy and paste the information from and to external applications using the right-click context menus shown.



Clear the **Years in Database** or the selected **Comments** using the same menu.

Note: If the workspace run period changes, editing a Year Mapping Group adjusts the run period to match the workspace run period. For years which overlap, the Year in Database and Comment are preserved. You are notified that the mapping group's run period has been adjusted. If you cancel out of the confirmation dialog the adjustment is not canceled (only the changes made after the dialog opened are canceled).

Note: In the input Database DMI specification, you should specify that the time range for each slot selection is Start Timestep to Finish Timestep. The Year Mapping Group will define the timestep, so it is only necessary to put in Start and Finish Timestep.



Invoking the Year Mapping Group

Invoke the Year Mapping group like any other group:

- **DMI**, then **Invoke** main menu item,
- the **Invoke** toolbar button,
- the DMI **Invoke** context menu item and
- the **Control-I** keyboard accelerator:

Year Mapping Algorithm

So what does this Year Mapping Group actually do? Following is the algorithm and description of errors.

```
For each DMI in the Year Mapping Group,
  For each Run Year, X
    Obtain the Year in the Database from the table, N
    Query the database for N to get the series of data
    Set the slot's values for year X with returned data from year N, dealing with
    leap days as shown in the following table.
  End For
End For
```

Table 1.1 describes how the mapping deals with Leap Years.

Table 1.1

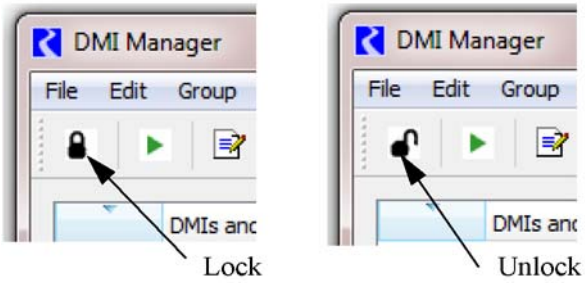
Year in Database	Run Year	Result
Non Leap Year	Non Leap Year	Bring in data exactly from database
Leap Year	Leap Year	Bring in data exactly from database
Leap Year	Non Leap Year	Remove the February 29th value
Non Leap Year	Leap Year	Repeat the February 28th value on February 29th

Following are some possible error conditions:

- If the workspace run period has changed and you have not opened the Year Mapping Group, invoking a Year Mapping Group results in an error. Open the Year Mapping Group and specify a map for the new run range.
- Executing an Output DMI in a Year Mapping Group leads to an error.
- Unspecified Years in the Database lead to an error.

Locking the DMI Manager Dialog

To lock the DMI manager dialog, select **Edit**, then **Lock** main menu item. Before locking the dialog you must close the DMI Parameter and DMI Editor dialogs (and will be warned if the dialogs are open).



When the dialog is locked, you cannot edit the DMI or DMI group configuration.

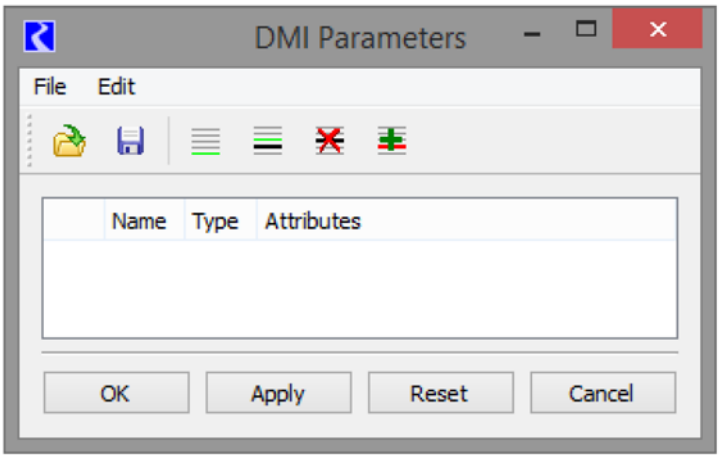
To unlock the DMI Manager dialog, select the **Edit**, then **Unlock** main menu item.

DMI Parameter Dialog

The DMI Parameter dialog enables the user to do the following:

- Create and delete parameters
- Edit a parameter's name
- Edit a parameter's type and attributes
- Order parameters

- Export and import parameters



Creating and Deleting Parameters

To create a parameter the user either:

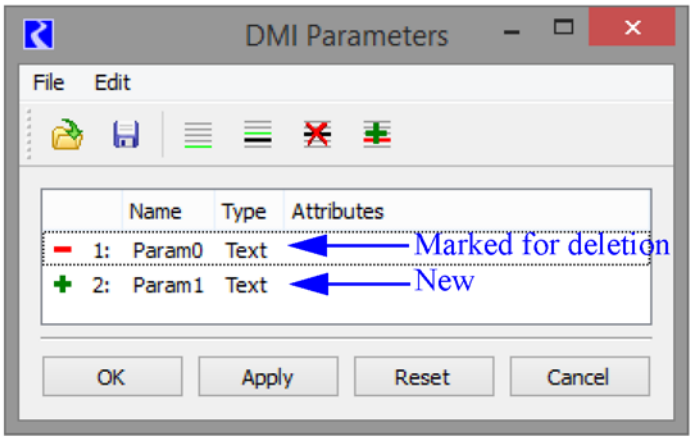
- Appends the new parameter to the end of the list (selects the **Edit**, then **Append Row** main menu item)
- Inserts the new parameter into the list, before the currently selected parameter (selects the **Edit**, then **Insert Row Before** main menu item)

The parameter has a default name, and + marks it as a new parameter. When the edits are applied the + mark is removed from the parameter.

To delete a parameter the user selects the **Edit**, then **Delete Row** main menu item. If the parameter is new (marked with +) it's removed from the list; otherwise − marks it for deletion. When the edits are applied parameters marked with − are removed from the list. See Figure 1.4.

To restore a row, the user selects the **Edit**, then **Undelete Row** main menu item; the − mark is removed from the parameter.

Figure 1.4



Editing a Parameter

When the user edits a parameter as described in the following sections ▲ marks the parameter as modified. When the edits are applied the ▲ mark is removed from the parameter.

Editing a Parameter Name

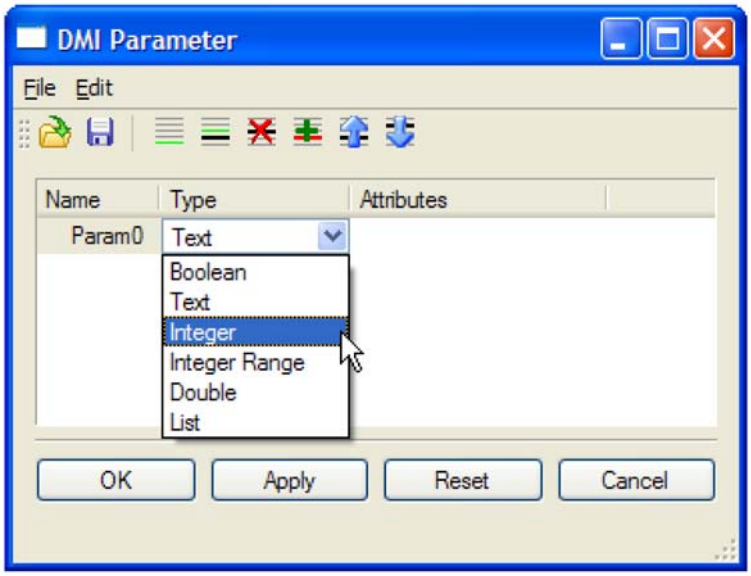
To edit a parameter name the user clicks twice in the **Name** cell, to edit the name in place. When the edits are applied the new parameter name is validated; the possible errors are:

- A blank name
- A name containing invalid characters (other than letters, digit and underscore)
- A duplicate name (the same name as another parameter)

Editing a Parameter Type and Attributes

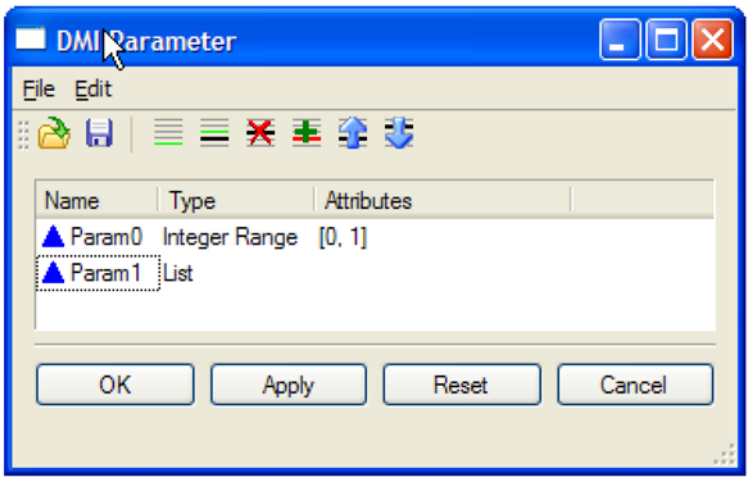
Type	Attributes	DMI Editor Control
Boolean	None	Menu containing “False” and “True”
Text	None	One line text editor, constrained to non-white space characters
Integer	None	Integer spinner, unbounded
Integer List	Minimum and maximum integer values (Default is 0, 1)	Integer spinner, bounded by minimum and maximum integer values
Double	None	One line text editor, constrained to syntactically valid double precision floating point values
List	List of text items, constrained to non-white space characters (Default is empty list)	Menu containing list of text items

To edit a parameter type, the user **Click-Pause-Clicks** in the **Type** cell, to activate the **Type** menu and then selects the parameter type.



If the selected type has attributes the default attributes are displayed in the **Attributes** column.

Note: A List parameter's default attribute is an empty list, so nothing is displayed.



To edit a parameter's attributes the user double-clicks the **Attributes** cell to activate the ... button, and then selects the ... button to open an attribute-specific dialog.

For an Integer Range the attribute-specific dialog enables the user to enter the minimum and maximum values, ensuring the maximum value is greater-than the minimum value:

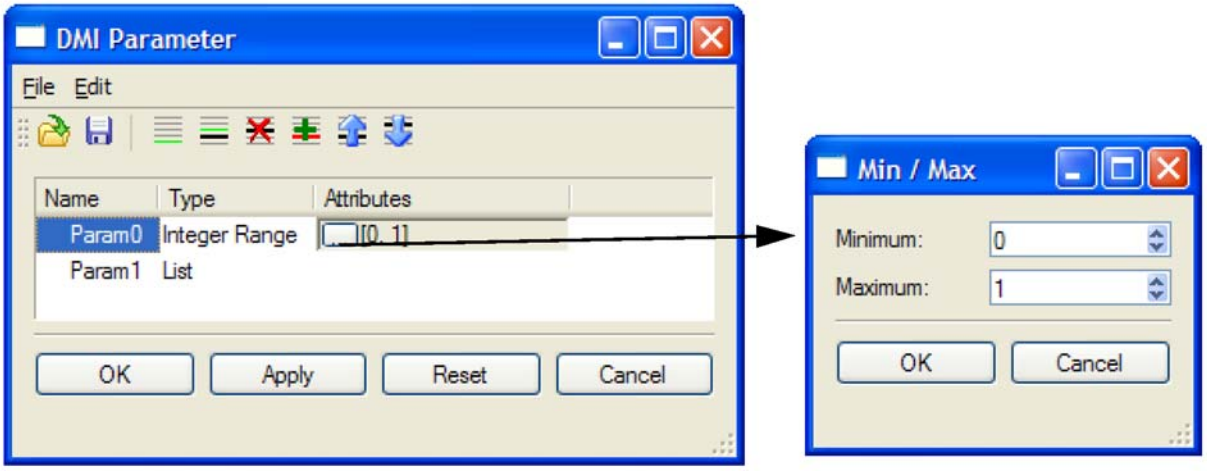


Figure 1.5 shows that for a List, the attribute-specific dialog enables the user to append, insert, delete, edit and order text items, ensuring the text items contain only non-white space characters (first image).

Figure 1.5

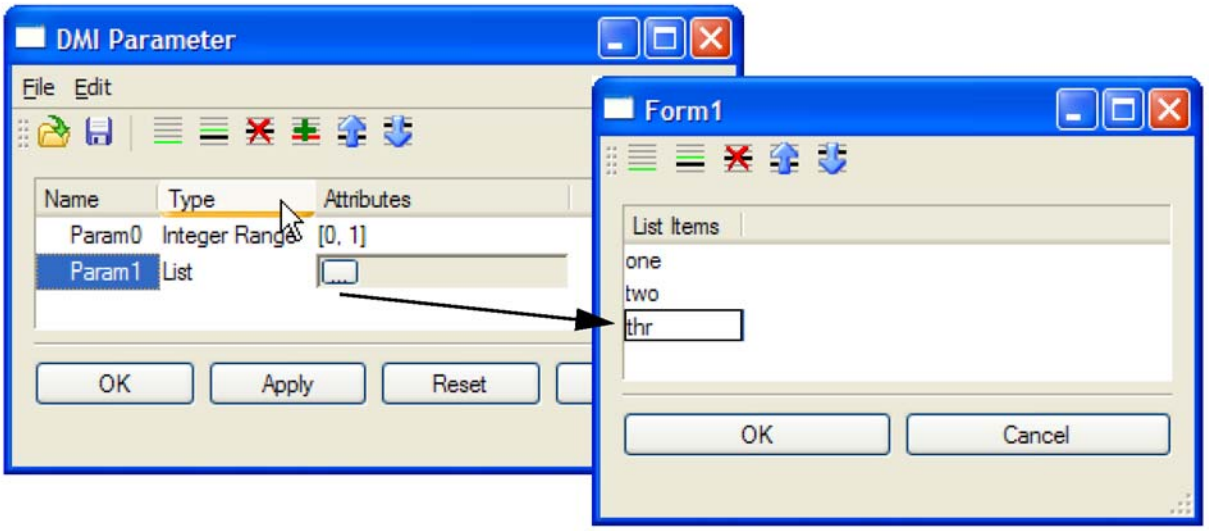
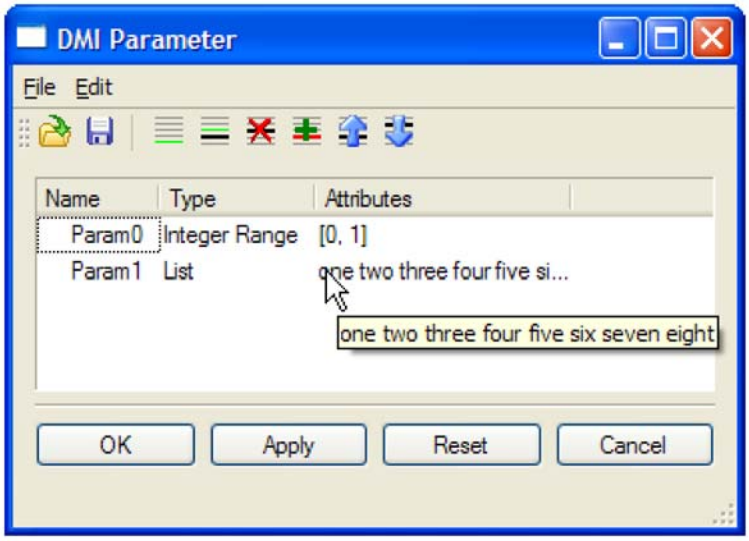


Figure 1.6 shows that if the resulting list contains more text than will fit in the **Attributes** cell, an ellipsis ("...") is displayed and the entire list is displayed in a tooltip when the mouse pointer is paused on the **Attributes** cell (second image).

Figure 1.6



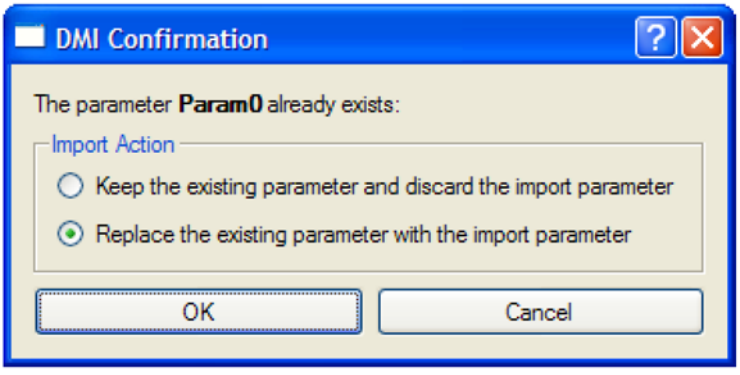
Exporting and Importing Parameters

Before exporting or importing parameters, all edits (+ , − or ▲) must be resolved (applied or canceled).

To export parameters the user selects the **File**, then **Export Parameters** main menu item, which opens a file chooser dialog. The user selects the file to export the parameters to, then selects **Save**. If the file has a ".gz" suffix the parameters are saved in a compressed format.

To import parameters the user selects the **File**, then **Import Parameters** main menu item, which opens a file chooser dialog. The user selects the file to import the parameters from, then selects **Open**.

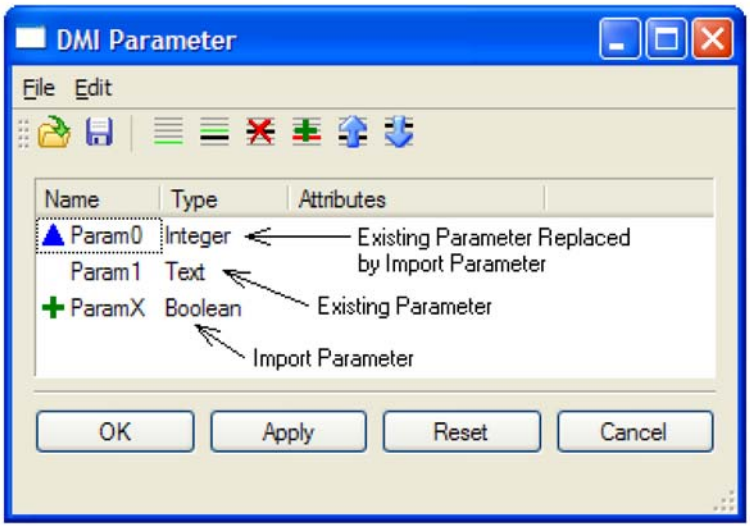
Parameters are imported by name; if the file contains a parameter which already exists, the user is asked whether to keep the existing parameter or replace it with the import parameter:



Also, parameters are imported as though they were edited. Import parameters which are new are marked with + ; import parameters which replace an existing parameter are marked with ▲ :

The user can then:

- Apply the import
- Cancel the import
- Further edit the parameters before applying the import and edits



Reconciling Parameter Edits

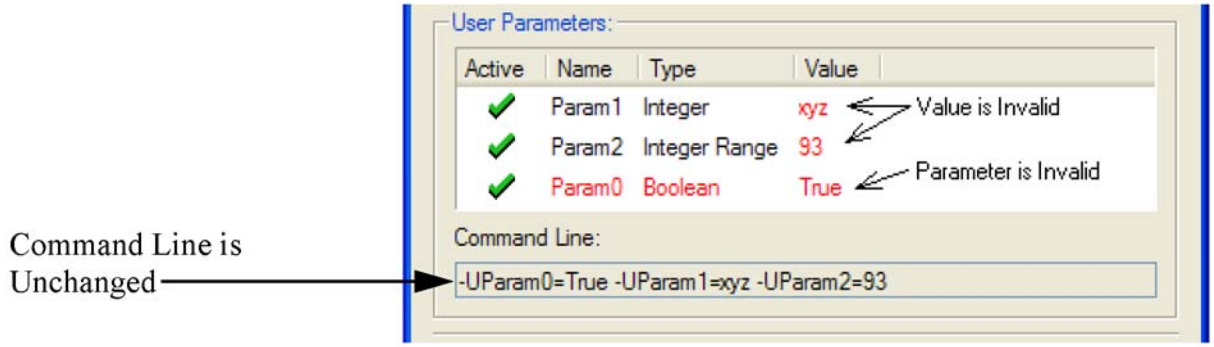
When the DMI Parameter dialog edits parameters it can invalidate DMIs using the parameters. Consider Param0 (a Boolean parameter), Param1 (a Text parameter) and Param2 (an Integer Range parameter):

- Deleting Param0 would invalidate a DMI using the parameter.
- Changing Param1's type from Text to Integer would invalidate a DMI using the parameter with a value of "xyz".
- Changing Param2's maximum value from 99 to 89 would invalidate a DMI using the parameter with a value of 93.

Although the DMI is invalid, it isn't changed; that is, when the DMI is invoked its executable's command line will still include:

```
-UParam0=True -UParam1=xyz -UParam2=93
```

However, when the user edits the DMI there will be a graphical indication that the DMI is invalid, and the user will be constrained in how the DMI can be edited.



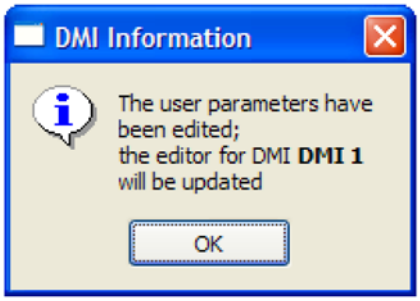
Param0 is shown entirely in *red* to indicate it is deleted; the user:

- Can deactivate the parameter
- Cannot edit the parameter's value

Param1's value and Param2's value are shown in *red* to indicate they are invalid; the user:

- Can deactivate the parameter
- Can edit the parameter, constrained to the parameter's new type or attributes. (In other words, Param1 can be edited as an integer, Param2 can be edited as an integer with a maximum value of 89.)

Finally, if the DMI Editor dialog is open when the user edits the parameters, a confirmation dialog appears, notifying the user that the DMI Editor dialog will be updated.

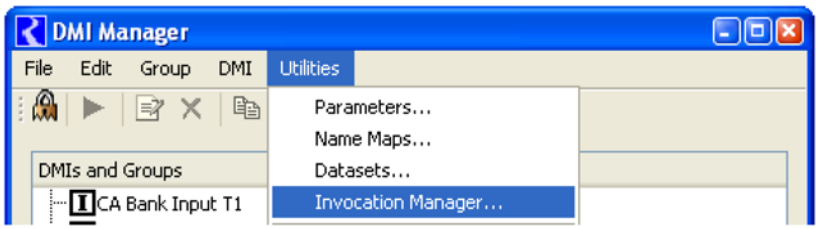


DMI Invocation Manager Dialog

The DMI Invocation Manager dialog displays all DMI invocations which RiverWare has recorded. From this dialog, the user can clear values that were set by a specific DMI.

> **Note:** DMI invocations are only recorded for DMIs that have been explicitly configured to do so. See "Record Invocations" on page 43 for details on Control File - Executable and Trace Directory DMIs. See "Creating a Database DMI" on page 58 for details on Database DMIs.

The dialog is opened from the DMI Manager dialog using the **Utilities**, then **Invocation Manage**r menu.



Overview

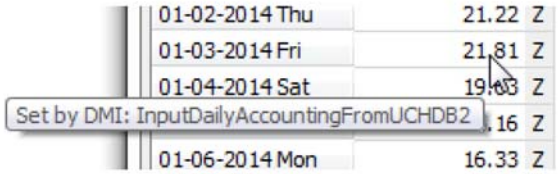
For each invocation the dialog displays the following:

- The DMI name.
- The date and time the DMI was invoked.
- The invocation status, one of:
  - "OK" - The invocation succeeded, without warnings.
  - "Warnings" - The invocation succeeded, with warnings.
  - "Errors" - The invocation failed, with errors.
- The user who invoked the DMI.
- The slots on which the DMI invocation set values and the number of values.

The DMI Invocation Manager is a singleton; that is, no other edits or views to any RiverWare window can be made while the DMI Invocation Manager is open. It must be closed first. Use the **File**, then **Close Window** menu, **Ctrl-W** shortcut or **Close** button.

Values set by a DMI with Invocations Recorded are given the "Z" flag. Tooltips on the slot window displays indicate which DMI set each values on the slot. For example, in Figure 1.7, on Jan 3, 2014 the value 21.81 was set by the InputDailyAccountingFromUCHDB2 DMI.

Figure 1.7

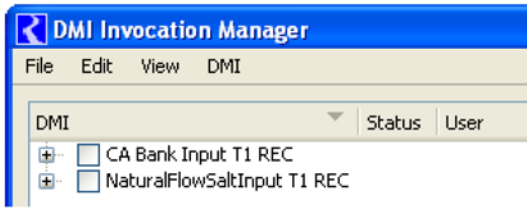


Views

The dialog has two views, **By DMI** and **By Date**.

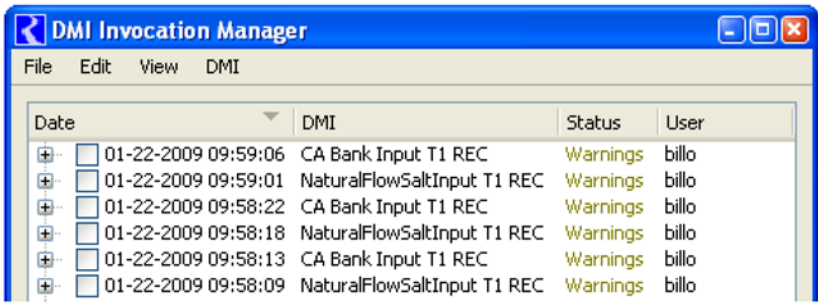


In the **By DMI** view, the invocations are grouped by DMI, with the invocations in reverse-chronological order (most recent is on the top).



In the **By Date** view, the invocations are displayed in a list, in reverse chronological order.

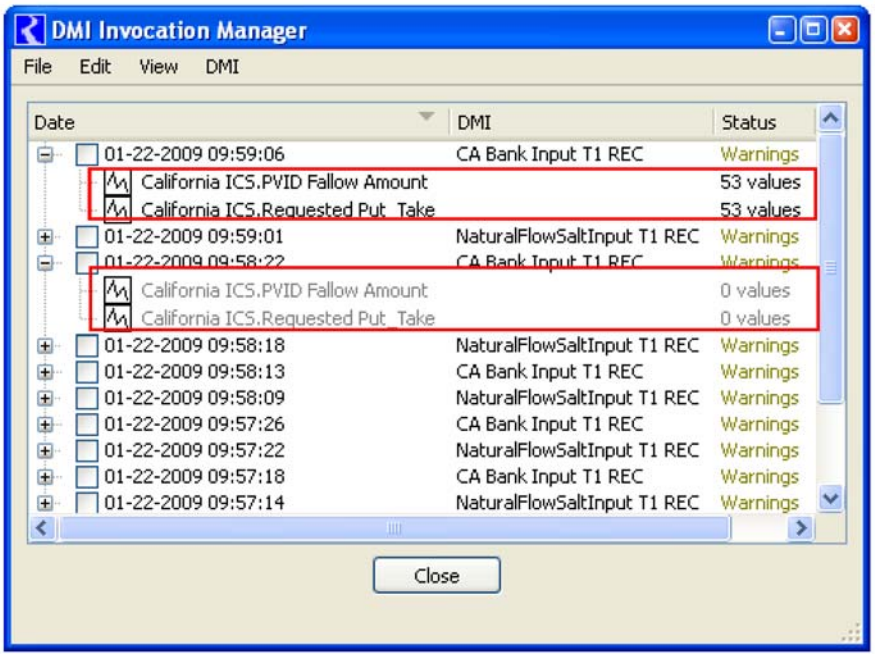
Note: The columns are sortable by selecting the column heading.



In both views, the slots set by an invocation are displayed in a tree view in alphabetical order.

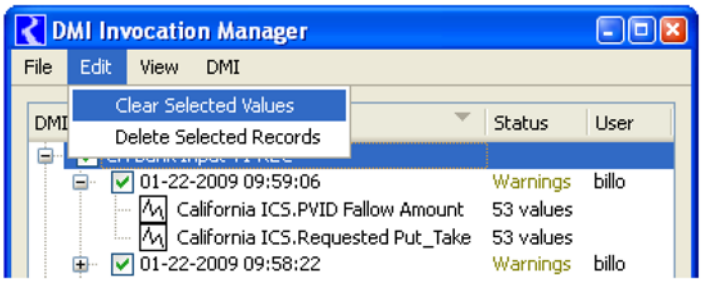
It is possible an invocation of a DMI will be superseded by other events, perhaps another invocation of the DMI or user inputs from the user interface. In Figure 1.8, the first invocation of “CA Bank Input T1 REC” (bottom) has been superseded by the second invocation (top), and consequently the first invocation (bottom) no longer is responsible for setting the values that are on the slot. This is indicated with the slot name grayed-out and “0 values” in the Status column.

Figure 1.8



Clearing Values

A user is able to select invocations, and then from the **Edit** menu **Clear** the values set by the invocation (i.e. the values in the slots).



The **Edit**, then **Clear Selected Values** feature clears values set by DMIs on a per-invocation basis; a user selects the DMI invocations to clear, and only the values set during those invocations will be cleared. Suppose a DMI is invoked twice, setting values on a SeriesSlot at timesteps t_1 and t_2 during the first invocation and at timesteps t_3 and t_4 during the second invocation. If a user selects the first invocation, only the values set at timesteps t_1 and t_2 will be cleared.

Following are some limitations on the clearing of DMI values:

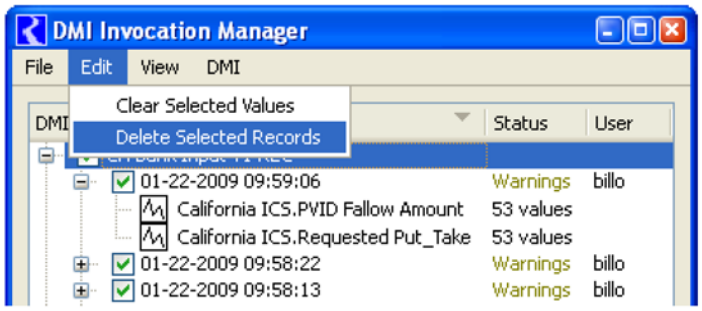
- Control File-Executable type DMIs which set values AND flags are not clearable.
- A user is able to clear values set by DMIs invoked from the user interface (the DMI manager dialog and the workspace menu) and from batch mode - not from RPL blocks or multiple run management.

- The feature clears values set by a DMI (as opposed to restoring values overwritten by a DMI). Specifically, if a DMI overwrites a non-NaN value, the clear feature will replace the value the DMI wrote with NaN - it will not restore the previous non-NaN value.
- The feature does not clear any “side effects” of a DMI. For example, if a DMI sets a Series Slot’s start date, the feature will not restore the previous start date.
- The feature clears values set by DMIs on series slots on a per-timestep basis. If a subsequent action (e.g., user input) overwrites a value set by a DMI, it will not be cleared.
- The feature clears table slots and scalar slots in their entirety. If a subsequent action (e.g., user input) overwrites a table value set by a DMI, it will be cleared.

DMIs can then be executed directly from the DMI Invocation Manager using the **DMI** menu.

Deleting Invocation Records

A user is able to select invocations, and then delete the invocation record by selecting **Deleted Selected Records** in the **Edit** menu. This operation deletes the record from the list of invocations but does not delete any of the values.



Invoking DMIs and DMI Groups

There sere five ways to invoke a DMI or DMI Group: from the **DMI Manager**, from the **Workspace**, from the **SCT**, from the **DMI Invocation Manager**, or from **RPL blocks**.

DMI Manager

To invoke a top-level DMI, select **DMI,** then **Invoke**; to invoke a DMI group, select **Group**, then **Invoke**.

Workspace

The Workspace **DMI** menu provides an alternate means of invoking DMIs and DMI groups - selecting the DMI or DMI group from the menu is equivalent to invoking the DMI or DMI group from the DMI Manager dialog (as described above). The user can alternate between a conventional menu (Figure 1.9) and a cascading menu (Figure 1.10). In the conventional menu the DMIs and DMI groups are sorted by name within type (first input DMIs then output DMIs then DMI groups).

Figure 1.9

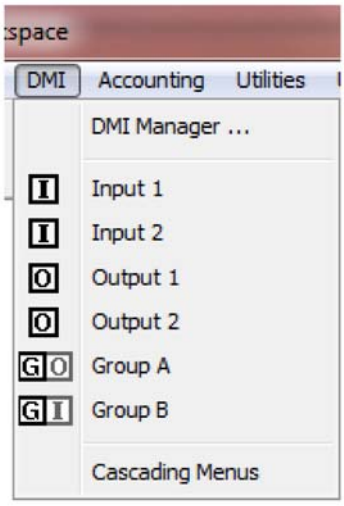
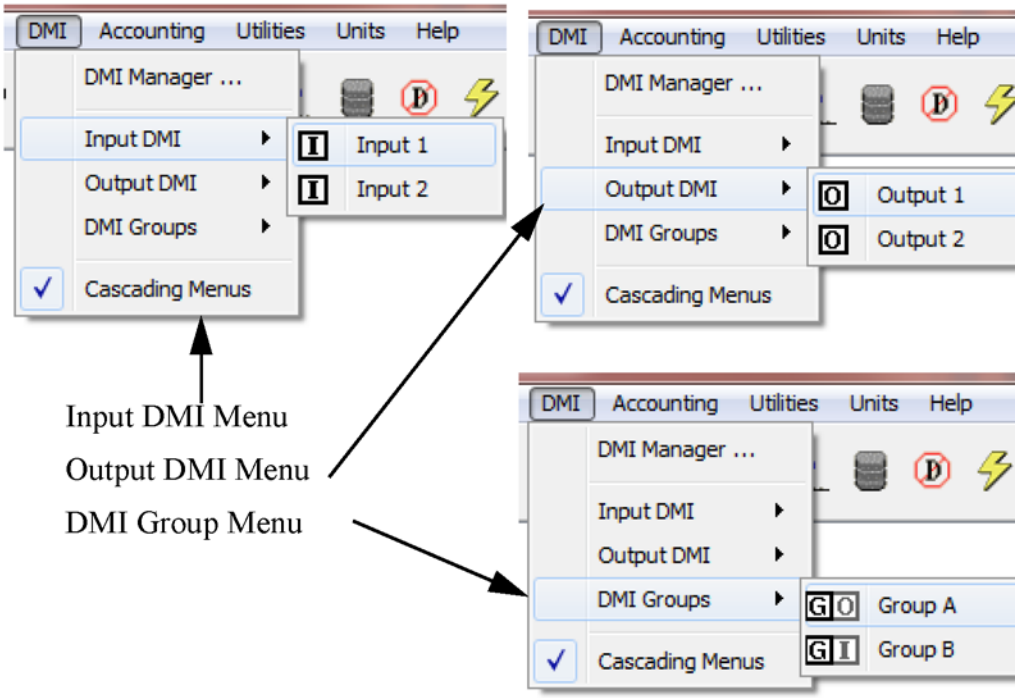


Figure 1.10



SCT

The SCT has the same **DMI** menu as the Workspace (described above).

DMI Invocation Manager

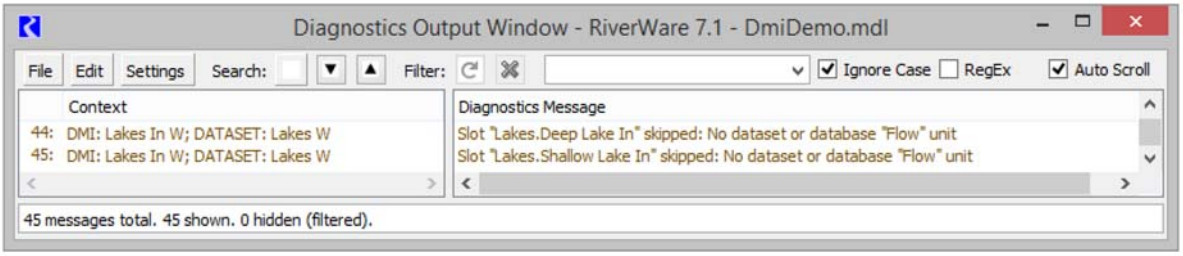
The DMI Invocation Manager has a similar menu to the Workspace **DMI** menu. The only difference is that there is no **DMI Manager** item on the DMI Invocation Manager. (The DMI Invocation Manager can only be accessed from the DMI Manager so closing the DMI Invocation Manager will go back to the DMI Manager).

RPL Blocks

A DMI or a DMI group can be executed from a RPL block (typically a rule) as either a pre-execution or post-execution. See "Executing DMIs From Blocks" in *RiverWare Policy Language (RPL)* for details.

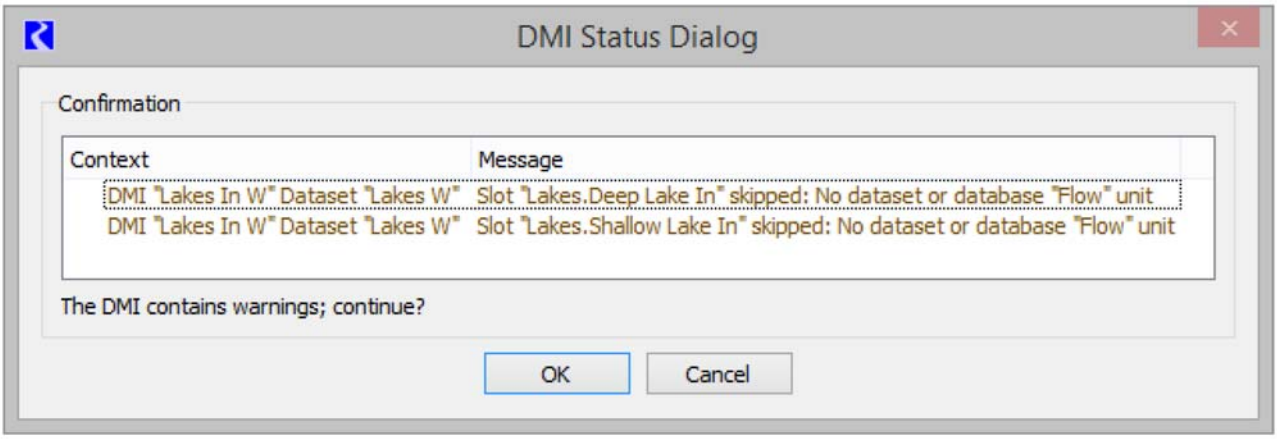
Warning and Error Messages

When a DMI is invoked, each warning is displayed in the Diagnostic Output dialog.



If the DMI is configured to Confirm Warnings (see "Name and Type" on page 61 and "Confirm Warnings Checkbox" on page 43) or an error occurs, a separate dialog called the DMI Status Dialog will open. Figure 1.11 shows the warning or error messages.

Figure 1.11



If the messages are just warnings and you wish to continue execution of the DMI (**Confirm Warnings** is selected), select **OK**. Otherwise, select cancel to stop execution.

Any errors generated by the DMI are displayed in the DMI Status Dialog and in the Diagnostic Output dialog in red text. After an error, the DMI stops executing.

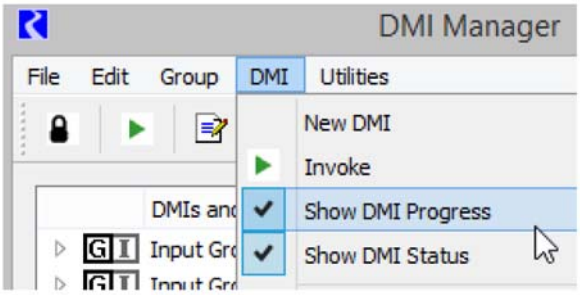
> **Note:** There are diagnostics available for DMIs. See "Workspace Diagnostics" in *Debugging and Analysis* for details.

DMI Status and Progress

The status of an executing DMI is shown in the bottom of the DMI manager. You can also configure to show a separate DMI Status Dialog that has optional panels for the Progress, Status, and Confirmation. This dialog shows

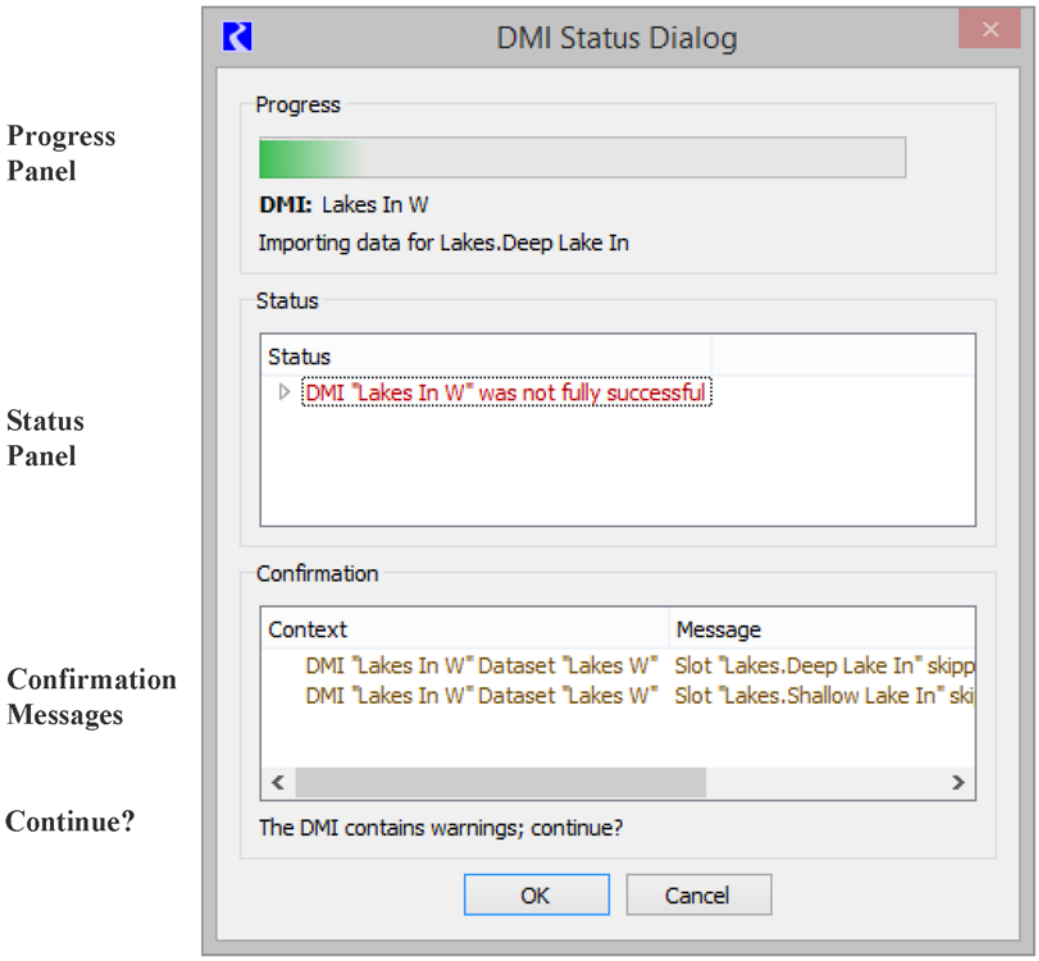
information on the DMI as it executes including the slots that it is processing and the location of the data being processed. To show this status dialog, from the DMI Manager, select **DMI**, then **Show DMI Status** or **DMI**, then **Show DMI Progress**. The selections are saved with the model file. In Figure 1.12, both **Show DMI Progress** and **Show DMI Status** are selected. In addition, there are Confirmation messages that must be confirmed before this DMI can continue.

Figure 1.12



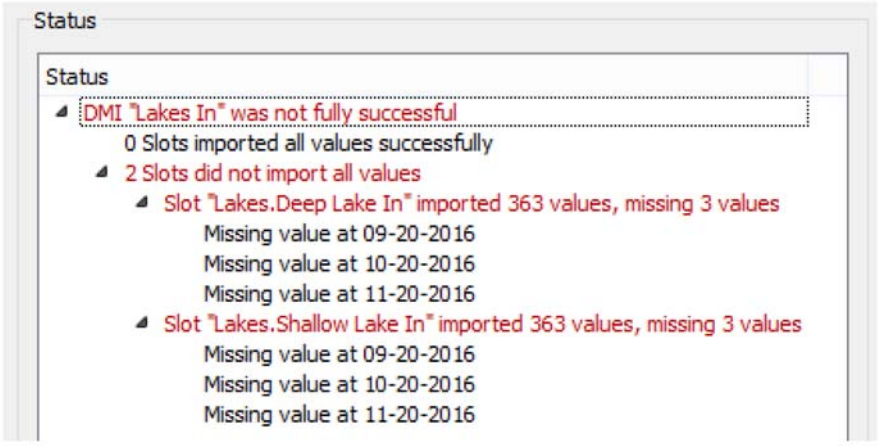
In Figure 1.13, the Progress panel shows that the DMI is executing including the Group (if there is one), the DMI, and the slots being processed.

Figure 1.13



The Status Panel shows the results (for an Input DMI only). For example, in Figure 1.14, the treeview has been expanded to show all of the information. This indicates that 363 values were imported for each slot, but three values in each slot were missing. The red messages are also copied to the Diagnostics Output Window so you can see the messages after closing the DMI Status dialog.

Figure 1.14



Icon Quick Reference

Table 1.2 DMI Manager dialog

Icon	Meaning
	Invoke the selected top-level DMI or DMI group
	Edit the selected top-level DMI
	Delete the selected top-level DMI or DMI group
	Copy the selected top-level DMI or DMI group
	Paste the previously selected top-level DMI or DMI group, creating a new top-level DMI or DMI group
	Add the selected top-level DMI to a DMI group
	Remove the selected member DMI from its DMI group
	Display the selected DMI's group membership
	Move the selected member DMI up within its DMI group
	Move the selected member DMI down within its DMI group
	Select to unlock the DMI Manager dialog
	Select to lock the DMI Manager dialog

Table 1.3 DMI Parameter dialog

Icon	Meaning
	Append a row to the list
	Insert a row into the list, before the selected row
	Delete the selected row
	Undelete the selected (and previously deleted) row
	Move the selected row up one position
	Move the selected row down one position
	Import parameters from a file
	Export parameters to a file
	Imported parameters which replace an existing parameter are marked with this icon
	Imported parameters which are new are marked with this icon

Table 1.4 Database DMI

Icon	Meaning
	Add a new Name Map
	Edit the selected Name Map or Dataset
	Copy the selected Name Map or Dataset
	Paste a copied Name Map or Dataset
	Delete or remove the selected Name Map or Dataset
	Choose a file using a file selector
	The name map, dataset or DMI has changes that have not been applied
	Indicates the name map, dataset, or DMI is locked by another dialog and cannot be edited.
	Indicates the name map, dataset, or DMI is scheduled for deletion and locked by another dialog and cannot be edited.

Chapter 2
Control File-executable DMI

The Data Management Interface (DMI) provides an efficient means of transferring large amounts of data between RiverWare and an external data source or sink. There are two directions in which a DMI can send data:

- An Input DMI, which transfers data from an external data source to RiverWare.
- An Output DMI, which transfers data from RiverWare to an external data sink.

Topics

- About the Control File-executable DMI, page 29
- Control File, page 30
- Data Files, page 37
- DMI Executable, page 40
- DMI Editor Dialog, page 42

About the Control File-executable DMI

The Control File-executable DMI consists of several components. Refer to Figure 2.1 for the relation between these components.

- The User-specified Control File is created by the user and contains a list of slots which the user wants imported or exported from the RiverWare model. The control file is an ASCII file which maps data between object.slots in the Workspace and data files
- The External DMI Executable is a separate software component which is called by the RiverWare DMI facility. For import DMIs, this external executable provides the data files which RiverWare loads into the model. For output DMIs, the external executable processes data files exported by RiverWare and performs such functions as saving the data in the database, reprocessing the data into reports, and preparing the data for use by statistical analysis programs.
- The RiverWare DMI Data Files are a set of files used to transfer the data to and from the RiverWare model. Each of these files contains the data for a single slot in the model. The files are created by either the External DMI Executable or RiverWare, depending on whether it is an input or an output DMI.

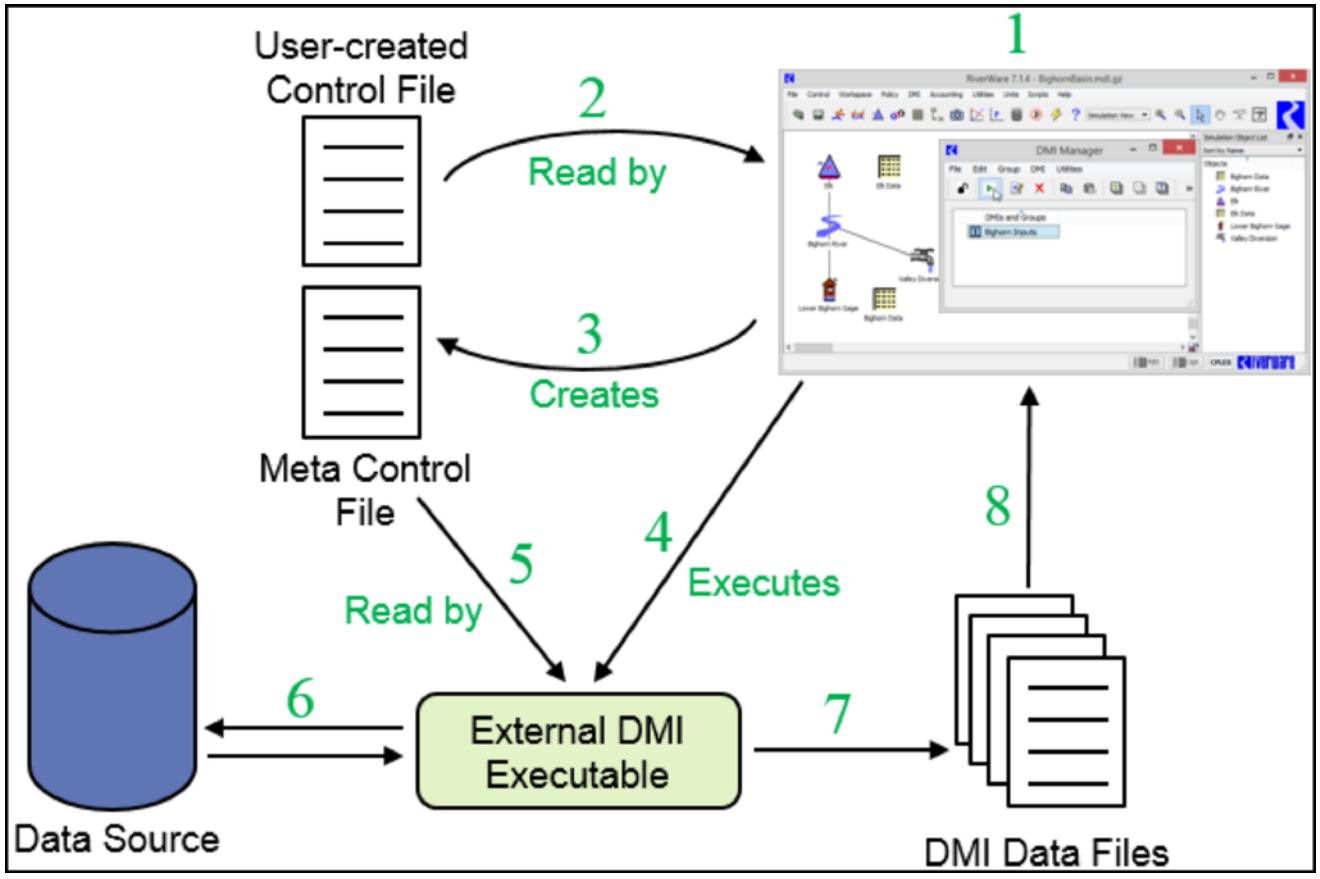
As shown in Figure 2.1, the user must minimally create a control file and an executable. Following is an example of how the DMI works. In this example we assume it is an Input DMI, but it is the same, just reversed for an Output DMI.

1. The user Invokes the Input DMI. RiverWare verifies the DMI to make sure it is valid.

2. RiverWare reads the user Control File and resolves all wild carding.
3. RiverWare writes the Meta Control File which has a fully specified line for each slot in the Control File.
4. RiverWare runs the External DMI Executable.
5. The executable reads the Meta Control File.
6. The executable interacts with the Data Source (i.e. the database) to retrieve the data.
7. The executable creates the DMI Data Files. .
8. Once the executable is complete, RiverWare reads the DMI Data Files and places data in the slots.

   Note: Alternatively, the user can create the DMI Data Files separately without using an executable. In this case steps 4-7 will be skipped

Figure 2.1 Control File - Executable Input DMI schematic



Control File

The control file is an ASCII file which the user provides. The purpose of the control file is as follows:

- Map between object.slots in the Workspace and data files. In the case of an import DMI, data is imported from the data files to the object.slots; in the case of an export DMI, data is exported from the object.slots to the data files.
- Provide information about the data files.

Each control file entry contains three pieces of information:

- An object.slot specification.
- One or more file names listed as file = name
- Optional keyword = value pairs.

The control file format is:

```
object.slot: file=name keyword=value keyword=value ...
```

If a file name contains spaces then it must be enclosed in double quotations marks (“), for example:

```
object.slot: file=”file name” keyword=value keyword=value ...
```

The DMI must also be configured to allow spaces in file paths; see “Allow Spaces in File Paths” on page 44 for details.

Object.Slot Specifications

The object.slot specifications indicate which slots will import from or export to data files. There are four ways of specifying the object:

- <object name> - A specific object name, e.g., “FtLoudoun,” which matches the object with that name. Aggregate object elements are specified as <aggregate name>:<element name>.
- <object type> - An object type, e.g., “LevelPowerReservoir,” which matches all objects of that type.
- <sub-basin name> - A sub-basin name, which matches all objects in the sub-basin.
- * - A wild card which matches all objects of all types.

There are three ways of specifying the slot:

- <slot name> - A specific slot name, e.g., “Inflow,” which matches the slot with that name.
- <slot type> - A slot type, e.g., “SeriesSlot,” which matches all slots of that type.
- dispatch - Any dispatching slots.

There are two ways of specifying a column within multi-column slots:

- <column name> - A specific column name, which matches one of the columns in the selected slot.
- no name - If no name is given, the DMI defaults to the first column.

An account or supply can be specified like so:

- <object name>^<account name>
- <supply name>.Supply - the “.Supply” portion of the string is necessary to tell RiverWare that this is a supply in the accounting system and not a slot in the physical system. If a DMI is used to bring in values, it should be

setting supplies in the accounting system rather than accounting slots. See "Setting Slots Versus Setting Supplies" in *Accounting* for details.

Also, the slots in a Slot Set can be specified using the SlotSet keyword, as follows. See "Slot Sets" in *User Interface*) for details.

```
SlotSet.<slot set name>
```

For example, SlotSet.Reservoir Inflows would expand to all of the slots in the Reservoir Inflows set.

The above slot specifications lead to many ways of specifying the object.slot pairs, as follows:

```
<object name>.<slot name>.<column name>
<object name>.<slot name>
<object name>.<slot type>.<column name>
<object name>.<slot type>
<object name>.dispatch.<column name>
<object name>.dispatch
<object type>.<slot name>.<column name>
<object type>.<slot name>
<object type>.<slot type>.<column name>
<object type>.<slot type>
<object type>.dispatch.<column name>
<object type>.dispatch
<sub-basin name>.<slot name>.<column name>
<sub-basin name>.<slot name>
<sub-basin name>.<slot type>.<column name>
<sub-basin name>.<slot type>
<sub-basin name>.dispatch.<column name>
<sub-basin name>.dispatch
*.<slot name>.<column name>
*.<slot name>
*.<slot type>.<column name>
*.<slot type>
*.dispatch.<column name>
*.dispatch
<object name>^<account name>
<supply name>.Supply
SlotSet.<Slot Set Name>
```

These pairs are listed from most exact to least exact. It is possible that more than one control file entry will select a specific object.slot pair in the Workspace. In this case, the more exact entry will always take precedence. For example, in the control file:

```
# All Confluence objects, Inflow1 slot.
Confluence.Inflow1: file=~/%o.inflow1.dat
# Confluence01 object, Inflow1 slot.
```

```
Confluence01.Inflow1: file=~/confluence01.inflow1.dat
```

Both entries will select Confluence01's Inflow1 slot (the first through an object type, the second through an object name). The more exact entry, the second entry, will take precedence.

There are two other concepts this example introduces: the control file may contain comments (lines beginning with #) and the file name may contain special characters (~ and %), described below.

Finally, note that object and slot names and types must be specified exactly as they appear in the user interface, including case and whitespace.

File Name and Directory Specification

The file name indicates from which data files the object.slots will be imported or to which data files the objects.slots will be exported. The file name may be either relative (beginning with `~/`) or absolute (beginning with `/`). If the file name is relative, then it is relative to the DMI's working directory. The DMI's working directory is where it creates its temporary files. If the `RIVERWARE_DMI_DIR` environment variable is set, the DMI's working directory is `$RIVERWARE_DMI_DIR/`*dmi_name*; otherwise, the DMI's working directory is `/tmp/`*dmi_name*. In both cases, *dmi_name* is the name that appears in the DMI Selector list.

The file name may contain %o and %s directives (currently, it may contain at most one of each). The %o is replaced with the object name, and the %s is replaced with the slot name. These are useful when using object types or slot types in the object.slot specifications.

> **Note:** Because dmi names, object names and slot names are mapped to directory and file names, characters which are not valid in directory and file names (currently all characters other than A-Za-z0-9/_.) are replaced with _ when the directory or file name is created.

Consider the following example:

```
RIVERWARE_DMI_DIR is /usr/local/riverware
```

The DMI name is “test dmi”

The control file contains the following entry:

```
Canal.Flow 1: file=~/%o.%s.dat
```

The file which the “Tellico Canal.Flow 1" object.slot would import from or export to would be:

```
/usr/local/riverware/test_dmi/Tellico_Canal.Flow_1.dat
```

The ~ has been replaced with the value of the working directory, RIVERWARE_DMI_DIR/<dmi name>, %o has been replaced with the object name, and %s has been replaced with the slot name (with the spaces having been replaced with _).

Multiple files can be specified on a single control file line entry for input dmi’s. This enables the user to import data into a slot from two or more files. Use of the data_date file keyword allows the user to specify where in the series slot each file should be placed. The syntax for multiple files is

```
Object.Slot: file=name file=name
```

The data files are imported from left to right. If the import = resize is specified, the slot will grow to accommodate the data. Overlapping data is allowed but values will be overwritten by the later data files. If the incoming data does not overlap and there are gaps, any untouched values that existed on the slot will remain.

Another control file available is `%tempdir`, which is replaced by a guaranteed unique temporary directory. When the DMI is invoked RiverWare generates the directory name, makes the directory and passes the directory to the DMI executable via the `RW_DMI_TEMPDIR` environment variable. The executable should be able to recognize `RW_DMI_TEMPDIR`. After the DMI finishes RiverWare removes the directory. The directory name will be similar to the following:

```
b05741ad-2b57-4f82-8044-bda67606e4c9
```

Because RiverWare removes the directory you most likely won't see these directories, but if you kill RiverWare or RiverWare crashes a directory might be left behind, in which case you can safely remove it.

Optional keyword=value Pairs

The optional keyword=value pairs provide additional information about the data files the slots will import. (The one exception to this is the file keyword, which identifies the data file itself.) Although the keywords are not case sensitive, the values can be.

The currently supported keywords are:

- file=<data file>

  The data file the slot will import from or export to, as described above.

- units=<units>

  The units of the data the slot will import or export. <units> must match exactly one of the units defined in the Units List dialog (see "Display Units" in *User Interface* for details on the Units List).

- scale=<scale> (float)

  The scale of the data the slot will import or export.

  Note: If either units or scale is not specified in the control file, then it defaults to the current setting in the user interface as specified by the active Unit Scheme; (see "Unit Schemes" in *User Interface* for details). Since the unit scheme can change from invocation to invocation, it is *strongly recommended* that the control file always contain units and scale (even if the scale is 1.0).

- precision=<display>|<model>

  The precision of the data the slot will import or export. The <display> precision is that specified in the Unit Scheme for the slot. The <model> precision is 17, the same precision as all internal calculations. The default is <display>.

- import=<fixed>|<resize>

  This keyword controls how the DMI sets the time series range of the series slot. The default is `fixed`.

  For either import option, the Begin Timestep of the slot is set in the same manner. If the data file contains a `start_date` header, the slot Begin Timestep is set to the `start_date` from the data file. If the data file does not contain a `start_date` header, the slot maintains its existing Begin Timestep.The data import always starts on the new Begin Timestep of the slot with the first value in the data file.

  For `import=resize`, if the data file contains an `end_date` header, the End Timestep of the time series range of the series slot will be set to the *later* of the `end_date` header in the data file or the number of timesteps necessary hold the number of data values in the data file given the updated Begin Timestep of the slot from above. If

the data file does not contain an `end_date` header, the End Timestep of the time series range of the slot will be extended, if necessary, to hold the number of data values in the data file given the updated Begin Timestep of the slot from above. The End Timestep of the slot will never be moved earlier if there is no `end_date` header in the data file.

For `import=fixed`, if the data file contains an `end_date` header, the End Timestep of the series slot is set to the `end_date` from the data file. Data will not be imported past the new End Timestep of the slot. If the data file does not contain an `end_date` header and the existing End Timestep on the slot is *later* than the updated Begin Timestep, the slot maintains its existing End Timestep. Data will not be imported past the End Timestep of the slot. If the existing End Timestep on the slot is equal to or earlier than the updated Begin Timestep, then the slot will be expanded to hold all of the data values in the data file starting with the updated Begin Timestep.

- flags=<true>|<false>

  The flags keyword specifies whether or not Series Slot flag values (Input, Target, Best Efficiency, Max Capacity, Drift, and Unit Values) will be exported or imported with the data. The syntax is `flags=true`; the default is <false>. If an input DMI, the keyword pairs in the data file should be specified as "`value flag`" with the flag being an integer value, e.g. Input = 1, Output = 2, etc.

- aggregate=<true>|<false>

  The aggregate keyword is used to specify whether or not the DMI should import/export all time series slots on an Aggregate Series slot or just the first time series column. If present, with the value "true", and if the slot indicated is an Aggregate Series slot, the data file is expected to (on import) or will (on export) contain data for all subslots (columns) in the Aggregate Series slot. If the keyword=value pair is absent, or if its value is "false", only the data for the first subslot (column) of the Aggregate Series slot are present.

  When Aggregate Series slot data are imported or exported in total, the data are in row-major format, one timestep per line. The subslots of the aggregation must be homogeneous in start-date, end-date and step size, and all subslots (columns) must be represented on each line.

  If used with the 'flags=true' keyword-value pair, the data will appear in the form:

  value flag value flag value flag...

  The flag is an integer number representing the flag. If the named slot is in fact the first subslot (column) of an Aggregate Series slot, the behavior is the same as if the subslot (column) name were absent.

The following three keywords are used to include text string metadata. They are mutually exclusive; you can only specify one at a time.

- slot_anno = <true>|<false>

  The slot_anno keyword indicates whether to import or export series slot notes; see "Exporting Notes Using DMIs" in *User Interface*for details. The default value for slot_anno is false, meaning no Notes will be written.

- slot_set_info = <true>|<false>

  The slot_set_info keyword indicates whether to export series slot value information. The default value for slot_set_info is false, meaning no information will be written.When true, the information corresponds to the tool tips seen on slot values on slot and SCT dialogs () will be written to the file; see "Series Slots" in *User Interface* for details. The string appears after the value. Examples of the information are "Set by DMI: Input DMI", "Set by Rule: (7) Flood Control".

- slot_opt_sol_info = <true>|<false>

The slot_opt_sol_info keyword indicates whether to export series slot optimization solution information . The default value for slot_opt_sol_info is false, meaning no information will be written. When true, the available per slot/value optimization solution information will be written to the file. The string appears after each value. Examples of the information are “Frozen at Lower Bound”, or “Frozen by (3) Minimum Load 47.6% between limits set by 3.1.1.1 and 2.1.1.1.”. See “Tooltips on Variables” in *Optimization* for details.

User keyword=value Pairs

The user keyword=value pairs provide an additional means by which the user can pass parameters to the DMI executable. The user keyword=value pairs are specified in the control file as:

```
!keyword=value
```

The keyword=value pairs are written to the metaControl file, with the ‘!’ removed, on all lines they apply to. The DMI executable can then use the keyword=value pairs to influence its behavior.

Example 2.1

```
#
# All LevelPowerReservoir Outflows
#
LevelPowerReservoir.Outflow: file=~/%o.Outflow units=cfs scale=1.0
#
# All LevelPowerReservoir Pool Elevations,
# but for Hiwassee use a different scale.
#
LevelPowerReservoir.Pool Elevation: file=~/%o.%s units=ft scale=1.0
Hiwassee.Pool Elevation: file=~/%o.%s units=ft scale=10.0
#
# All LevelPowerReservoir Storages.
#
LevelPowerReservoir.Storage: file=~/%o.%s units=cfs-day scale=1.0
#
# All LevelPowerReservoir Hydrologic Inflows.
#
LevelPowerReservoir.Hydrologic Inflow: file=~/%o.%s units=cfs scale=1.0
#
# Finally, all SeriesSlots which are not selected by one of
# the preceding lines. (Remember that *.<slot type> is the
# least exact specification.) The file ~/start_end is a data
# file which contains only the start_date: and end_date:
# attributes. This will serve to change the start date and
# end date for all series slots which import this file.
#
*.SeriesSlot: file=~/start_end
```

Data Files

The data files are the files from which the slots will be imported or to which the slots will be exported. They contain both the data attributes and data values. The attributes are specified as keyword:value pairs, and all attributes must come before the first data value. Although the keywords are not case sensitive, the value in the keyword:value pair can be.

The currently supported keywords are:

- The start date of a series slot:

  start_date: yyyy-mm-dd hh:mm

  start_date: 1996-04-15 06:00

- The end date of a series slot:

  end_date: yyyy-mm-dd hh:mm

  end_date: 1996-04-22 06:00

- The start date of where the data should be imported.

  data_date: yyyy-mm-dd hh:mm

  data_date: 1996-10-22 06:00

  data_date specifies where in the series slot the data is to be imported. Implicitly, data_date is only useful with input DMIs. It was designed for use with the multiple data file feature in control files: Object.Slot: file=... file=... in which data files are imported from left to right. If all data files were imported at the beginning of the series slot then values from later data files would overwrite values from earlier data files. Therefore, all data files (except possibly the first) should contain the data_date keyword, to import their values at the correct location in the series slot and to prevent their values from overwriting values from earlier data files.

- The timestep of the data being exported from a series slot:

  timestep: <count> <timestep>

  <count> is an integral value.

  <timestep> is one of HOUR, DAY WEEK, MONTH, YEAR.

  timestep: 6 HOUR

- The units of the data in the data file.

  units: units

  units: acre-feet/month

  For multi-column table slots:

  units: <column 0 units> <column 1 units> ...

  units: ft cfs

- The scale of the data in the data file.

  scale: scale

scale: 100

For multi-column table slots:

scale: <column 0 scale> <column 1 scale> ...

scale: 1 1000

- The new display units for the slot importing the data. Also, if the units keyword is not specified, then the units of the data being imported is the new_units:

  set_units: new_units

  set_units: cms

- The new scale or the slot importing the data. Also, if the scale keyword is not specified, the scale of the data being imported is the new_scale.

  set_scale: new_scale

  set_scale: 10

- The new Lower Bound for the slot. In the units and scale specified on the slot.

  set_min: new_min

  set_min: 10

- The new Upper Bound for the slot. In the units and scale specified on the slot.

  set_max: new_max

  set_max: 1000

- The new Optimization Min Value for the table slot. In the units and scale specified on the slot.

  opt_min: <column 0 min> <column 1 min> ...

  opt_min: 10 10

- The new Optimization Max Value for the table slot. In the units and scale specified on the slot.

  opt_max: <column 0 max> <column 1 max> ...

  opt_max: 10000 100000

- The new display format of the slot importing the data.

  display_format: <new_format>

  <new_format> is one of float, integer, scientific.

  display_format: scientific

- The new display precision of the slot importing the data.

  display_precision: new_precision

  display_precision: 4

- The new period of the periodic slot importing the data.

  periodic_period: <count> <timestep>

<count> <timestep> is one of 6 HOUR, 12 HOUR, 1 DAY, 1 MONTH, N YEAR where N is 1 to 99.

periodic_period: 2 YEAR

- The new base year for the multi-year periodic slot importing the data.

  periodic_base: new_year

  periodic_base: 1998

- The new interval type for the periodic slot importing the data.

  periodic_interval: <new_interval_type>

  <new_interval_type> is one of regular, irregular.

  periodic_interval: irregular

- The new interval size for the regular periodic slot importing the data.

  periodic_regular_size: <count> <timestep>

  <count> <timestep> is one of 1 HOUR, 6 HOUR, 12 HOUR, 1 DAY, 1 MONTH.

  periodic_regular_size: 1 MONTH

- The new data interpolation method for the periodic slot importing the data.

  periodic_data_interp: <new_method>

  <new_method> is one of lookup, interpolate.

  periodic_data_interp: lookup

- The new header units for a periodic slot with numeric column headers that is importing data.

  column_units: new_units

  column_units: cfs

- The new header scale for a periodic slot with numeric column headers that is importing data.

  column_scale: new_scale

  column_scale: 10

- New symbolic row dates for the irregular periodic slot that is importing the data.

  periodic_rows: <new_row_dates>

  <new_row_dates> is a space separated list of <symbolic_date> entries.

  <symbolic_date> takes the form of minute/hour/day/month/multi-year index, where each piece is represented by an index number (minute (0), hour (0-23), day (1-31), month (1-12), year (1-99)). Symbolic dates need only be complete to the longest appropriate interval for the period of the particular slot. For example, 0/3 is 3:00, 0/3/21 is 3:00 Day 21, 0/3/21/4 is 3:00 April 21, and 0/3/21/4/2 is 3:00 April 21, Year 2.

  periodic_rows: 0/0/16/2 0/0/2/3 0/0/16/6 0/0/2/10 0/0/16/12

- New column header numbers for a periodic slot with numeric column headers that is importing data.

  periodic_columns: <new_column_entries>

<new_column_entries> is a space separated list of numeric values.

periodic_columns: 1000 4000 10000 40000 100000

The data values are specified either one per line (for series, list, and scalar slots) or one row per line (for table and periodic slots). For table slots, the row may include a new row label before the data values. If the label contains spaces, it must be enclosed in double quotes. As with the control file, the data files may contain comments (lines beginning with #).

Series slot notes (see "Notes on Series Slots" in *User Interface*) can also be represented in data files for import or export. They are specified as:

```
<value> anno {NoteGroup} {Note}
```

where <value> will be a numeric value, "anno" is a keyword, NoteGroup is the name of the NoteGroup and Note is the actual note. The "<" and ">" will not exist in the file, while the "{" and "}" do exist, delimiting the Note Group name from the note text itself. For Example, a line in the file might be:

```
102.031 anno {Storm Events} {4.3 Inches}
```

where Storm Events defines the Note Group to which the note belongs and 4.3 Inches is the note itself.

DMI Executable

The DMI executable is invoked by RiverWare, which then waits for it to exit. For an import DMI, the executable creates data files from an external data source. For an export DMI, the executable populates an external data sink from data files.

Nearly any programming language can be used to write the DMI executable. Often the choice of tools will be based on user experience and existing tools that can communicate with the database.

> **Note:** A Windows batch (.bat) script could work as a DMI executable. It does need to include an exit statement: `exit /b 0` The convention for a DMI executable is a zero exit status indicates success, a non-zero exit status indicates an error. Without the exit statement the bat script would return the "DMI executable failed: Exited with 1." error.

The DMI executable is passed the following command line arguments:

Meta-control File

The path to the meta-control file. The meta-control file is a copy of the control file described above, with wildcards resolved. Each line is of the form:

```
object.slot: file=<file name> [optional keyword=value pairs]
```

The DMI executable should use the object.slot specification to determine for which objects and slots it must provide data (import DMI), or for which objects and slots data is being provided (export DMI). The executable should use the file specification to determine to where to write the data to be read by RiverWare (import DMI), or from where to read the data exported by RiverWare (export DMI).

Working Directory

The working directory is the DMI's working directory, as described above. It is an artifact of an earlier implementation and can safely be ignored by the DMI executable.

Start Data (YYYY-MM-DD)

The start date of the data the executable must provide (import DMI), or of the data which is being provided (export DMI).

Start Time (HH:MM)

The start time of the data the executable must provide (import DMI), or of the data which is being provided (export DMI).

End Data (YYYY-MM-DD)

The end date of the data the executable must provide (import DMI), or of the data which is being provided (export DMI).

End Time (HH:MM)

The end time of the data the executable must provide (import DMI), or of the data which is being provided (export DMI).

Timestep (6HOUR)

The timestep of the data the executable must provide (import DMI), or of the data which is being provided (export DMI).

System Parameters

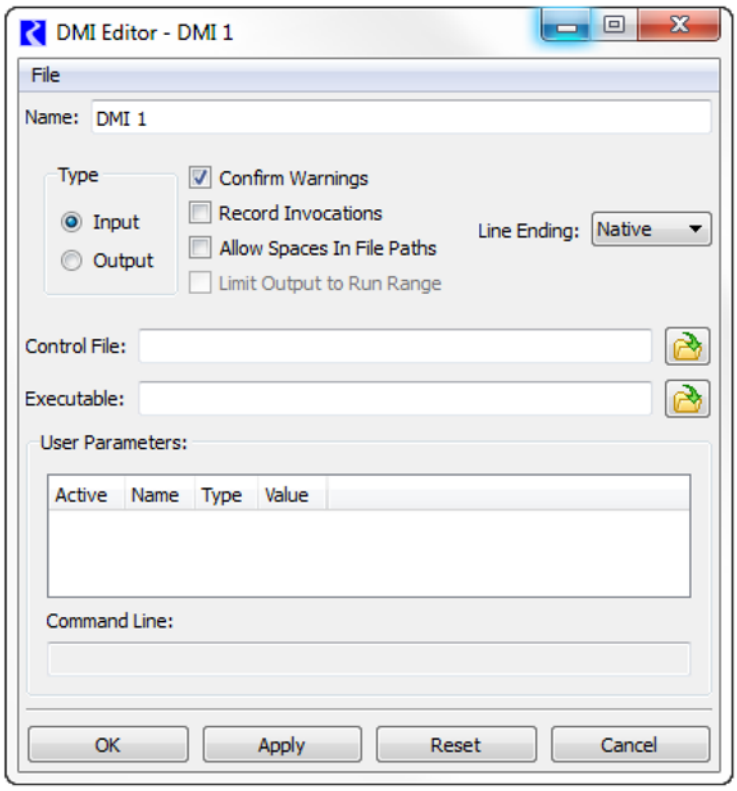
Not implemented.

User Parameters

The user parameters, in the form: -Ukeyword=value The intention of the user parameters is described above.

DMI Editor Dialog

The DMI Editor dialog enables the user to edit the DMI name, type, control file path, executable file path and other configuration options.



Name

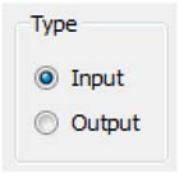
To edit the DMI name, the user selects the **Name** field to edit the name in place:

When the edits are applied the new DMI name is validated; the possible errors are:

- A blank name
- A duplicate name (the same name as another DMI)

Type

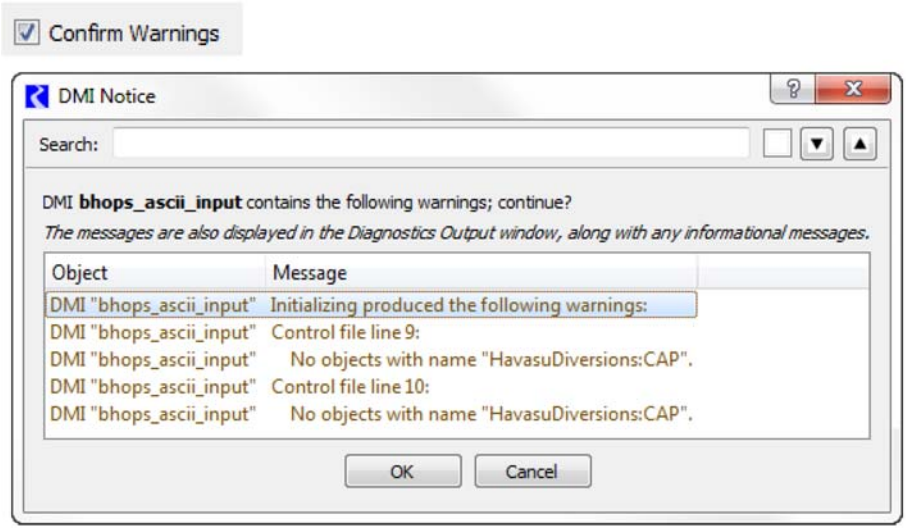
To edit the DMI type, the user selects the appropriate Type.



Confirm Warnings Checkbox

The **Confirm Warnings** check box controls whether you wish for the DMI to continue when warnings are generated. If **Confirm Warnings** is selected, the user is prompted to confirm any warning messages that appear; if not checked, the DMI continues after any warnings. Figure 2.2 is a sample dialog that appears during execution when the **Confirm Warnings** check box is selected.

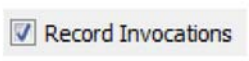
Figure 2.2



Record Invocations

The **Record Invocations** check box allows RiverWare to maintain information which chronicles input DMI invocations. With this information, values set by DMIs can be cleared on a per-invocation basis.

If a user wishes to clear all values that are input by a DMI, this box should be checked. If a user knows values set by a particular DMI won't be cleared then it is not necessary to maintain information about the DMIs invocations and this box should remain unchecked. (The information occupies memory and could, over time, degrade performance.)



The check box is only enabled for input DMIs and the default is unchecked. See "DMI Invocation Manager Dialog" on page 18 for details on clearing values set by an input DMI.

> **Note:** When the **Record Invocations** box is checked, series values imported by the input DMI are given the "Z" flag indicating they were set by a DMI.

Allow Spaces in File Paths

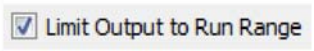
The **Allow Spaces in File Paths** allows the user to specify paths that have spaces. Without this box checked, the DMI would replace spaces in paths with '_'.

> **Note:** If users have written their DMI executable to recognize white space as the end of a path, and they begin using spaces in path names, then the executable should be modified.



Limit Output to Run Range

The **Limit Output to Run Range** toggle allows the user to specify that output data should only be written for the run range (Start Timestep to Finish Timestep). Without this box checked, the DMI will write the entire series of data. This configuration is only available for Output DMIs.



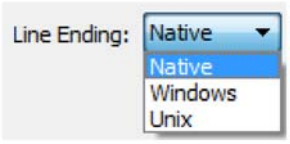
Line Ending

The **Line Ending** menu allows the user to specify the type of line ending sequence to write for output DMIs. This option is useful when you are running RiverWare on one platform, e.g. Windows, but your output data processing utilities and scripts are on another platform, e.g. Solaris.

The choices are:

- Native (default, uses the format for the current platform)
- Windows (CR NL)
- Unix (NL)

CR = carriage return, NL = newline



Control File

To edit the DMI control file, the user either:

- Selects  to open a file chooser dialog and selects the control file
- Selects the **Control File** field to edit the control file in place



The control file can contain environment variable references of the form:

- `$VARIABLE`
- `$(VARIABLE)`
- `${VARIABLE}`

VARIABLE is a letter followed by zero or more letters, digits or underscores. If the variable is followed by a letter, digit or underscore it must be "quoted" with () or {}. For example:

- `$DMI_DIR/Control/Res.ctl`
- `$(DMI_DIR)_$TARGET/Control/Res.ctl`

When the edits are applied the control file is validated; the possible errors are:

- A blank control file

All other validation occurs when the DMI is invoked, including:

- Replacing environment variable references with their values
- Verifying the resulting path refers to a readable file

Executable

To edit the DMI executable file, the user either:

- Selects  to open a file chooser dialog and select the executable file
- Selects the **Executable** field to edit the executable file in place



> **Note:** Alternatively, the user can create the DMI Data Files separately without using an executable. In this case the executable field can be left blank.

The executable file can contain environment variable references, the same as the control file.

All validation occurs when the DMI is invoked, including:

- Replacing environment variable references with their values
- Verifying the resulting path refers to an executable file

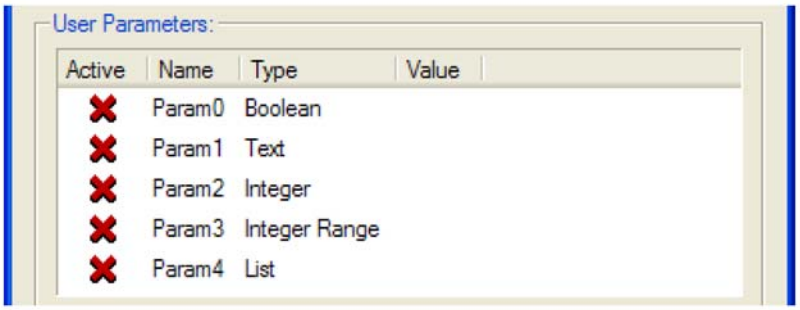
User Parameters

To edit the DMI user parameters the user specifies:

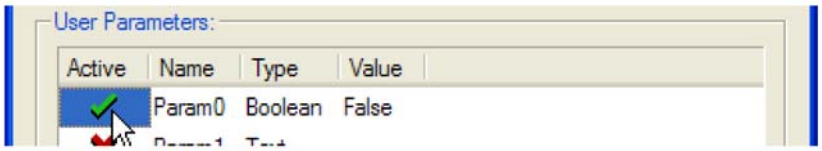
- If a parameter is active (e.g. if it should be included on the DMI executable's command line)
- If a parameter is active, its value

✖ indicates an inactive parameter; ✔ indicates an active parameter; by default parameters are inactive:

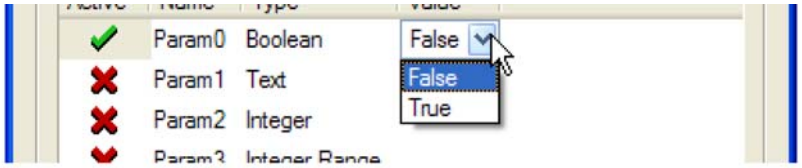
The user toggles a parameter's active state by selecting the **Active** cell; when a parameter is active its value is displayed in the **Value** cell.



The user edits a parameter value by clicking twice in the **Value** cell, activating a control appropriate for the parameter type.

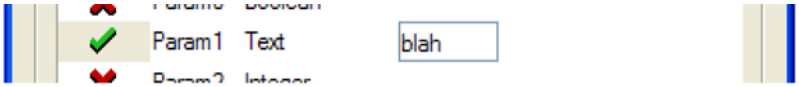


For a Boolean parameter the control is a menu containing False and True.

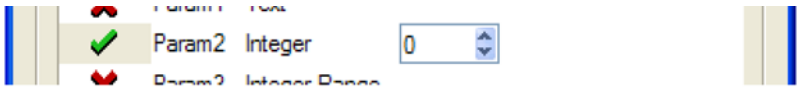


For a Text parameter the control is a one line text editor. When the edits are applied the parameter's value is validated; the possible errors are as follows:

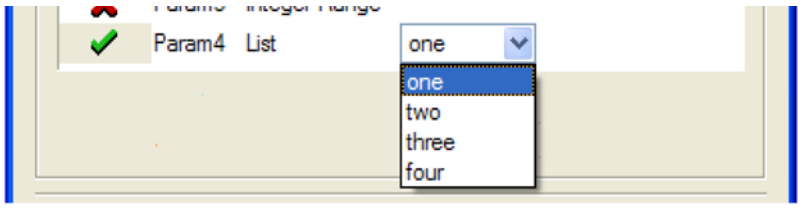
- The value is blank
- The value contains white space



For an Integer parameter the control is an unbounded integer spinner. For an Integer Range parameter the control is an integer spinner, bounded by the parameter's minimum and maximum values.

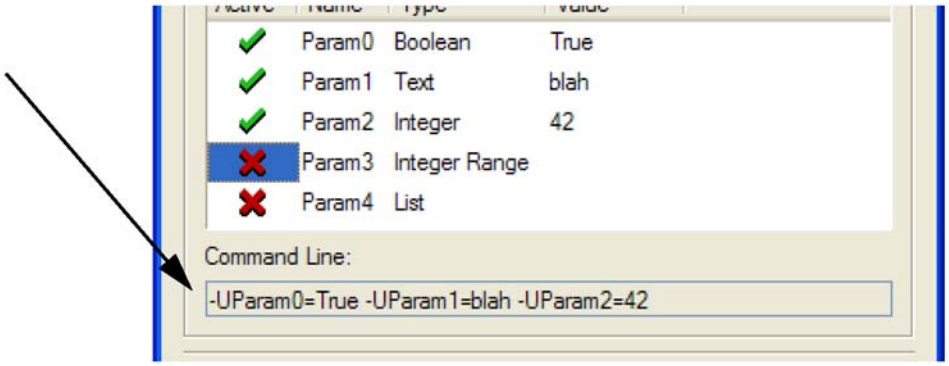


For a List parameter the control is a menu containing the list items defined in the DMI Parameter dialog.



Command Line

As the user edits the user parameters they are displayed as they will appear on the DMI executable's command line:



The command line is not editable, but it can be horizontally scrolled, copied-and-pasted and dragged-and-dropped (to a Microsoft Word document, for example).

Chapter 2
Control File-executable DMI

Chapter 3
Trace Directory DMI

The Trace Directory Data Management Interface (DMI) is similar to the Control File-Executable approach where the user develops a *control file* that specifies the slot data to move into or out of RiverWare. However, instead of an executable, the user specifies a top directory, under which the data files for traces are accessed in subdirectories named with the trace number.

Topics

- About the Trace Directory DMI, page 49
- Trace Directory DMI Editor Dialog, page 51

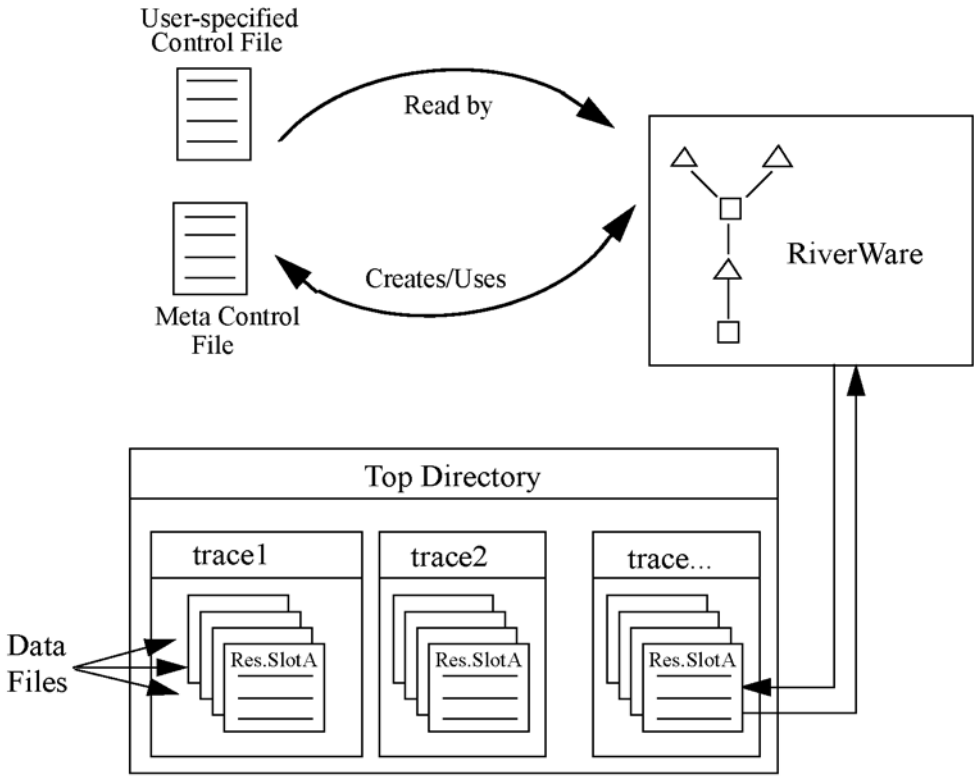
About the Trace Directory DMI

The Trace Directory DMI provides a means to transfer data between RiverWare and data files in a known directory structure. There are two directions in which the DMI can transfer data:

- An Input DMI transfers data from data files into RiverWare.
- An Output DMI transfers data from RiverWare into data files.

The Trace Directory DMI consists of several components. Figure 3.1 illustrates the relationships among these components.

Figure 3.1 Trace Directory DMI components

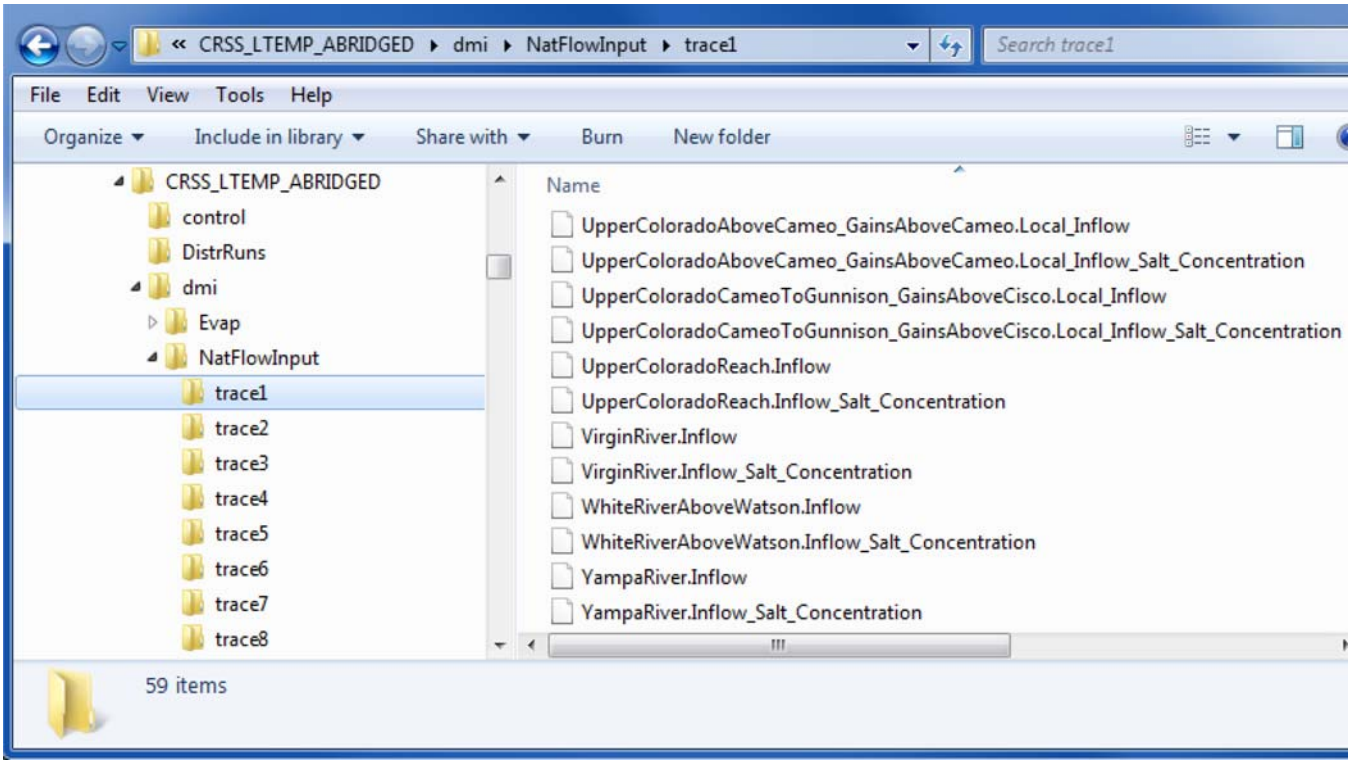


- The user-specified Control File is created by the user and contains a list of slots which the user wants imported or exported from the RiverWare model. The control file is an ASCII file which can map data between object.slot names in the Workspace and data file names. See "Control File" on page 30 for details. But no `file=` needs to be specified as the Trace Directory DMI can resolve the file name from the trace directory path and the object.slot name. The control file could be as simple as the following entry:

  ```
  BigRes.Outflow: units=cfs scale=1
  ```

- The user-specified top directory is a directory under which data files will be located within subdirectories named with the trace number corresponding to the data (i.e. trace1, trace2, etc.) Multiple traces could be associated with input or output DMIs from multiple runs, but running a Trace Directory DMI outside of MRM would use a single trace folder. Figure 3.2 is an example of a top directory (NatFlowInput) and trace subdirectories, each of which contains slot data files.

Figure 3.2



- The RiverWare DMI data files are a set of files used to transfer the data to and from the RiverWare model. Each of these files contains the data for a single slot in the model. In an output DMI RiverWare will create the files, or will read existing files for an input DMI. See "Data Files" on page 37 for details.

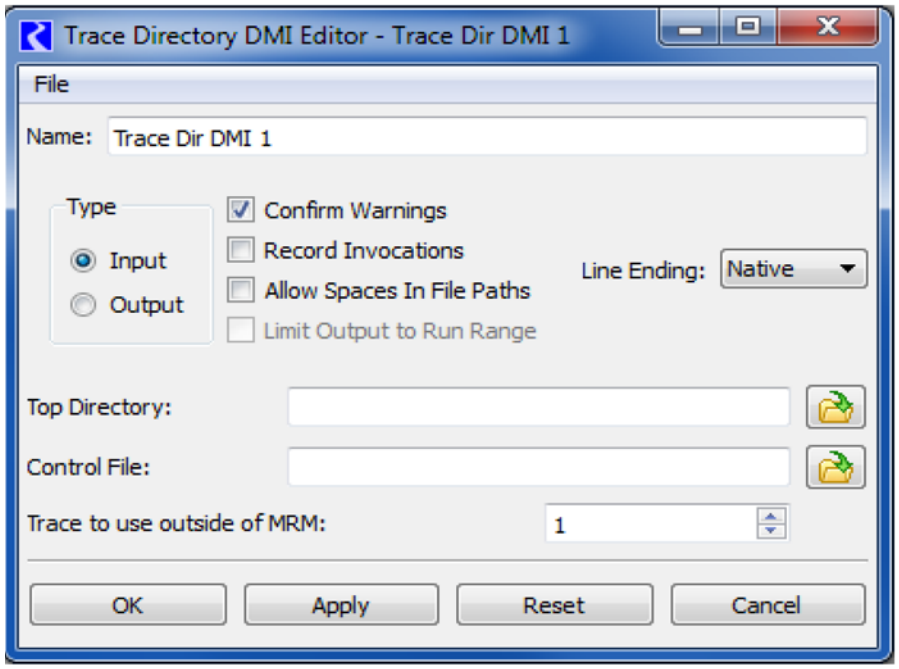
Following is an example of how the DMI works. In this example we assume it is an Input DMI, but it is the same, just reversed for an Output DMI.

1. The user Invokes the Input DMI.
2. RiverWare verifies the DMI to make sure it is valid.
3. RiverWare reads the user Control File and resolves all wild carding. It then substitutes the directory path for the current trace into the file keyword path for each entry in the control file. If a file keyword is not provided, it generates one from the trace directory path and the object.slot name. It then writes the Meta Control File, which has a fully specified line for each slot in the Control File.
4. RiverWare then uses the Meta Control File to read the DMI Data Files from their appropriate directory and imports the data into the slots in RiverWare.

Trace Directory DMI Editor Dialog

A new Trace Directory DMI is created from the main DMI Manager dialog by selecting **DMI**, then **New DMI**, then **Trace Directory DMI**. Edit the DMI by double-clicking in the DMI Manager or by selecting **Edit** from the **Edit** menu; the Trace Directory DMI Editor dialog appears.

Figure 3.3



Many of the configuration options are the same as those for the Control File-Executable DMI. The following list references the discussions in that section for the equivalent controls:

- **Name -** Name, page 42.
- **Type -** Type, page 43.
- **Confirm Warnings Toggle -** Confirm Warnings Checkbox, page 43.
- **Record Invocations -** Record Invocations, page 43.
- **Allow Spaces in File Paths -** Allow Spaces in File Paths, page 44.
- **Limit Output to Run Range -** Limit Output to Run Range, page 44.
- **Line Ending -** Line Ending, page 44.
- **Control File -** Control File, page 45.

To specify a Top Directory and the Control File the user either:

- The  button to open a chooser dialog and select the directory/file.
- Select the field to edit the path in place.

The top directory path can contain environment variable references of the form:

- `$VARIABLE`
- `$(VARIABLE)`
- `${VARIABLE}`

VARIABLE is a letter followed by zero or more letters, digits or underscores. If the variable is followed by a letter, digit or underscore it must be “quoted” with () or {}. For example:

- `$DMI_DIR/EvapData/`
- `$(DMI_DIR)_$TARGET/EvapData/`

The **Trace to use outside of MRM** control is an integer spinner where a trace number can be selected or typed. If this DMI is executed outside of a multiple run, this trace number is used to determine the subdirectory under the top directory to use for reading or writing data. For example, if the trace number is 3, the subdirectory “trace3” is used as the location for the DMI input or output files. Within MRM, the trace number is automatically substituted.

Chapter 3
Trace Directory DMI

Chapter 4
Database DMI

The Database DMI provides a direct connection between RiverWare and an external database. Databases currently supported are the U.S. Army Corp of Engineers Hydrologic Engineering Center's Data Storage System (DSS), the Hydrologic Database (HDB), and Microsoft Excel.

Following are the slot types that can be read or written by each type of DMI:

- The DSS interface supports series data and paired data (Table Slots with two columns).
- The HDB interface supports series slot data.
- The Excel interface support series, periodic, table, and scalar data.

This section presents an overview of the Database DMI, then presents details on creating database DMIs, datasets, name maps, and sharing database DMIs.

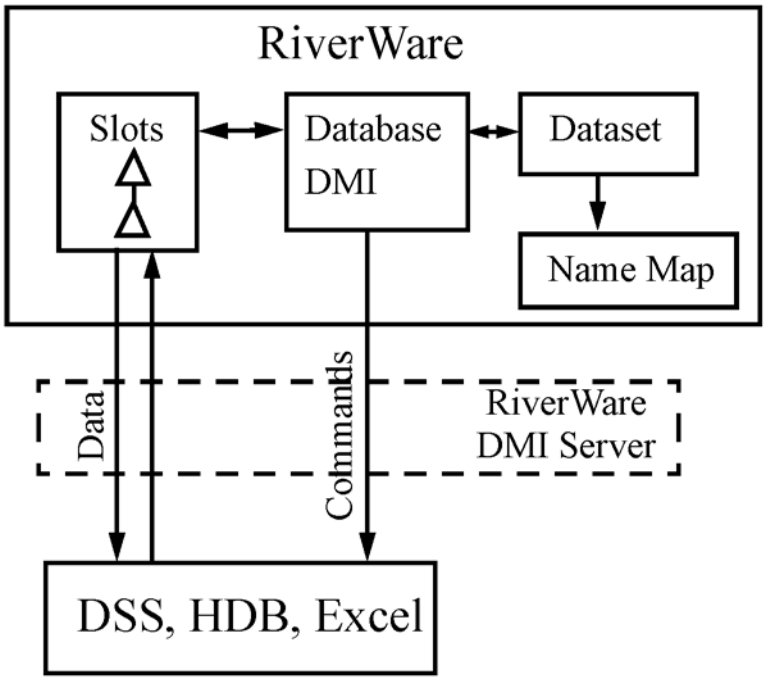
Topics

- About the Database DMI, page 55
- Creating a Database DMI, page 58
- Datasets Overview, page 82
- DSS Datasets, page 83
- HDB Datasets, page 86
- Excel Datasets, page 100
- Name Mapping, page 117
- Sharing Database DMIs, page 120

About the Database DMI

The Database DMI allows RiverWare to import and export data directly from and to the database. In this document, a *database* refers to a specific database implementation, for example a DSS file or Excel workbook. Figure 4.1 shows the interaction between RiverWare, the DMI Server and the database. The user sets up the Name Maps, configures the Datasets, and identifies slots and time intervals at which to import/export data in the Database DMI.

Figure 4.1



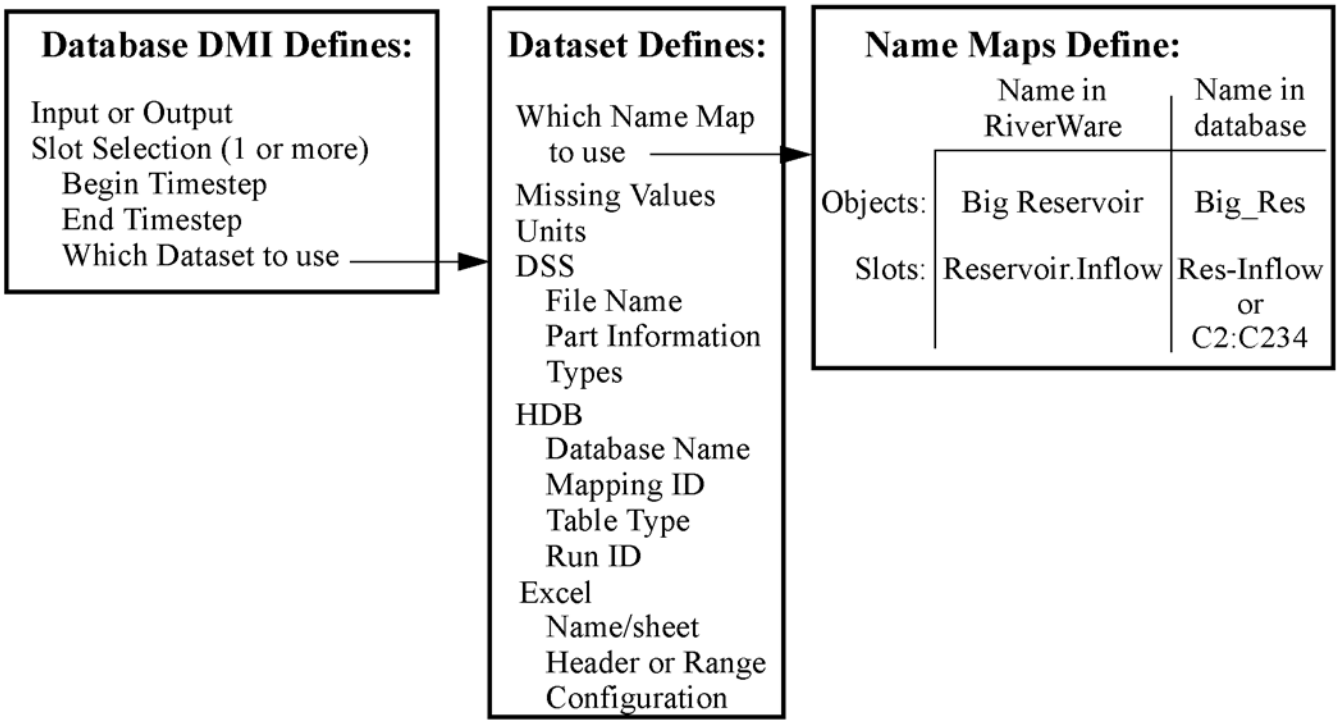
When the Database DMI is invoked, the Dataset and Name Maps are read specifying the database to use, the units and what slots/objects are named in the database. Any wildcarded slot selections are then fully specified and then any Name Maps are applied. Then commands are passed to the DMI server: for an input DMI, the server locates the data in the database and imports data into RiverWare slots; for an output DMI, the server takes the RiverWare slot data and inserts it into the appropriate place in the database.

> **Note:** The DMI server is a separate application included with the RiverWare installation package. It is not visible to the user and no further discussion is presented.

Figure 4.2 shows in more detail the three pieces of a database DMI and how they work together to define a DMI.

> **Note:** With the DMI Database Editor you can configure an entire Database DMI in one dialog but these three components still exist and interact with one another. See "Creating a Database DMI" on page 58 for details.

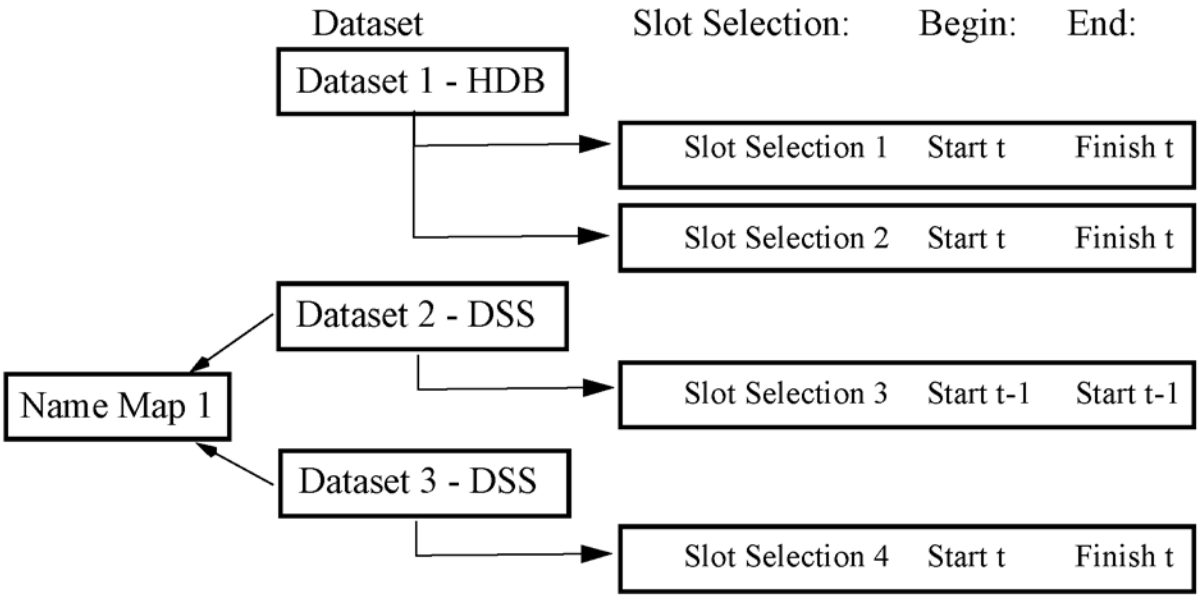
Figure 4.2



The three pieces (slot selections, datasets, and name maps) were designed to be very flexible. In the simple case of importing or exporting 1 slot, the user needs 1 slot selection, 1 dataset, and optionally 1 name map (DSS only). It is possible to configure the system with more functionality which leads to more complexity.

For example, in Figure 4.3, slot selections 1 and 2 share dataset 1 which is an HDB dataset; no Name Map is required. In this DMI, Slot selection1 is an input from HDB for the full run and then slot selection 2 is an input from HDB for the full run. Slot selection 3 is used to input initial conditions and uses Dataset 2 which points to a separate DSS file in Dataset 2. Slot selection 4 inputs from a DSS file referenced in Dataset 3 for the full run. Both Dataset 2 and 3 are DSS datasets and share Name Map 1.

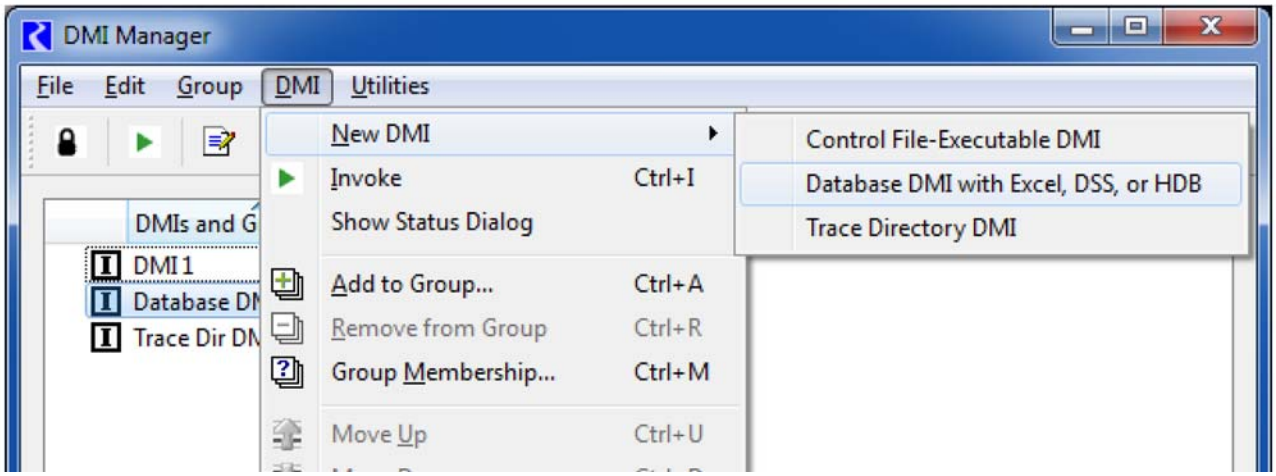
Figure 4.3



We will now look at each component starting with creating a Database DMI, then look at Datasets followed by Name Maps.

Creating a Database DMI

A Database DMI is created from the main DMI Manager dialog using the **DMI**, then **New DMI**, then **Database DMI with Excel, DSS, or HDB** menu.



There are two ways to define and edit a Database DMI:

- **Define a Name Map, Dataset and then use the Database DMI editor:** You create a Name Map, then a Dataset, then a Database DMI that pulls it all together.

- **Use the Database DMI Editor:** The Database DMI Editor allows you to create a fully functional DSS Database DMI from a single dialog and provides access to other dialogs for Excel and HDB DMIs. It fully expands the slot selection to show the exact configuration that will be used. For example, it displays that BigReservoir.Inflow will be imported using BigRes Res-Inflow and displays the part information and units that will be used.

Once a new Database DMI is added to the DMI manager, the user can edit it using the Database DMI Editor. The Database DMI Editor dialog provides a convenient place to create, edit, and view database DMIs. It also provides a single dialog for configuration of a fully functional DSS Database DMIs. To that end, it allows the user to specify all required configuration information and as much optional configuration information as is feasible. The real power of this dialog is that the user can see the fully specified information that will be sent to the database, which is especially useful for DSS.

To open the New Database DMI Editor, the user selects **Edit**, then **Edit with New Database DMI Editor**. The Database DMI dialog appears.

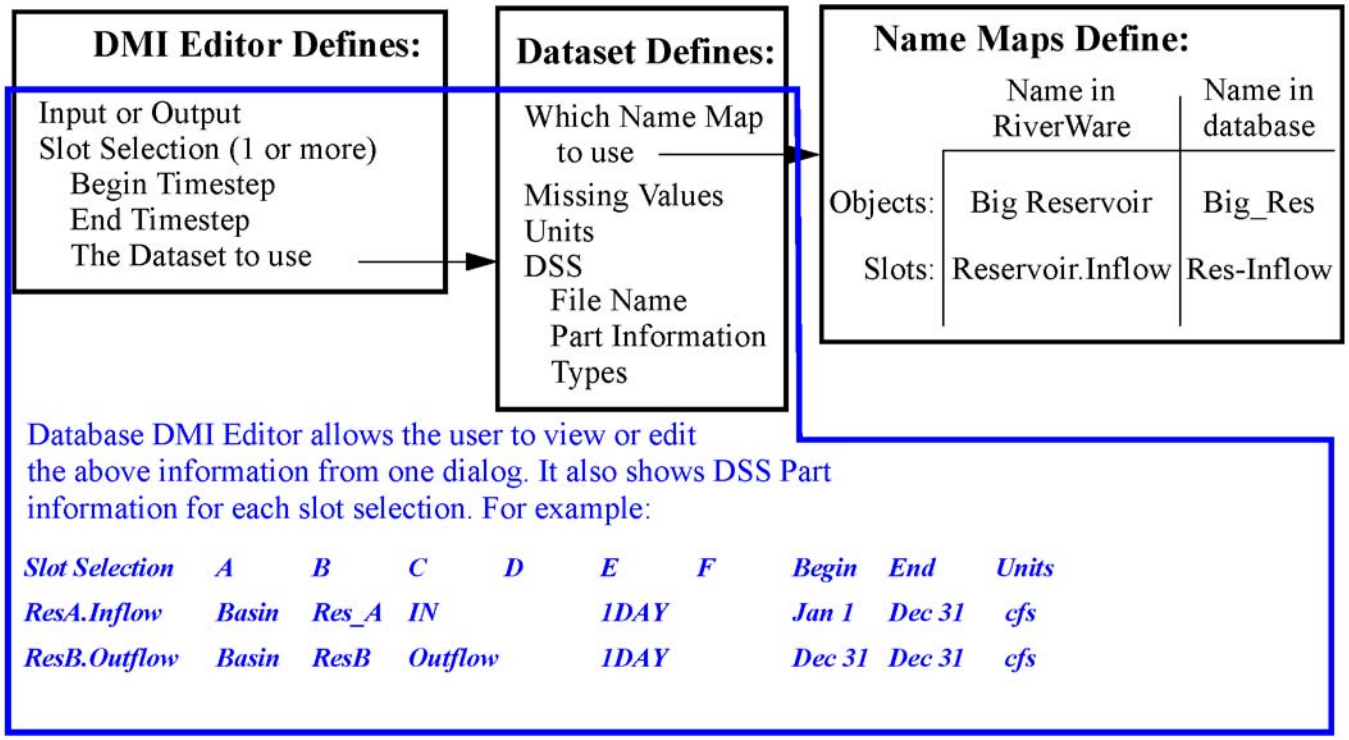
This dialog is like an editable view into the Dataset Manager. Thus, the user can see what has already been configured in those dialogs but also make edits that will be applied to those pieces. It is important to note that if the user edits a dataset or Database DMI in the editor window, the user cannot also edit the individual components at the same time.

> **Note:** Because there are many approaches to create and edit Database DMIs, there are multiple ways to edit the same information, e.g. you could edit a dataset from both the dataset manager and the Database DMI Editor. To avoid conflicts, the dialog that first starts editing a name map or dataset creates a "lock" on the information and no other dialog can edit that feature. When a lock has been created, the lock icon  is added to any other dialogs that are trying to edit that information. If you see this icon, the information is not editable from that location. Similarly, if a name map or dataset has been scheduled for removal from another dialog, the "delete lock" icon is shown. .

Figure 4.4 shows a similar figure to the one presented in "About the Database DMI" on page 55, but with the Database DMI Editor also shown.

> **Note:** The information in each of the components is represented in the dialog, but in a slightly different organization. The parts of the dialog are described in detail in the following section.

Figure 4.4



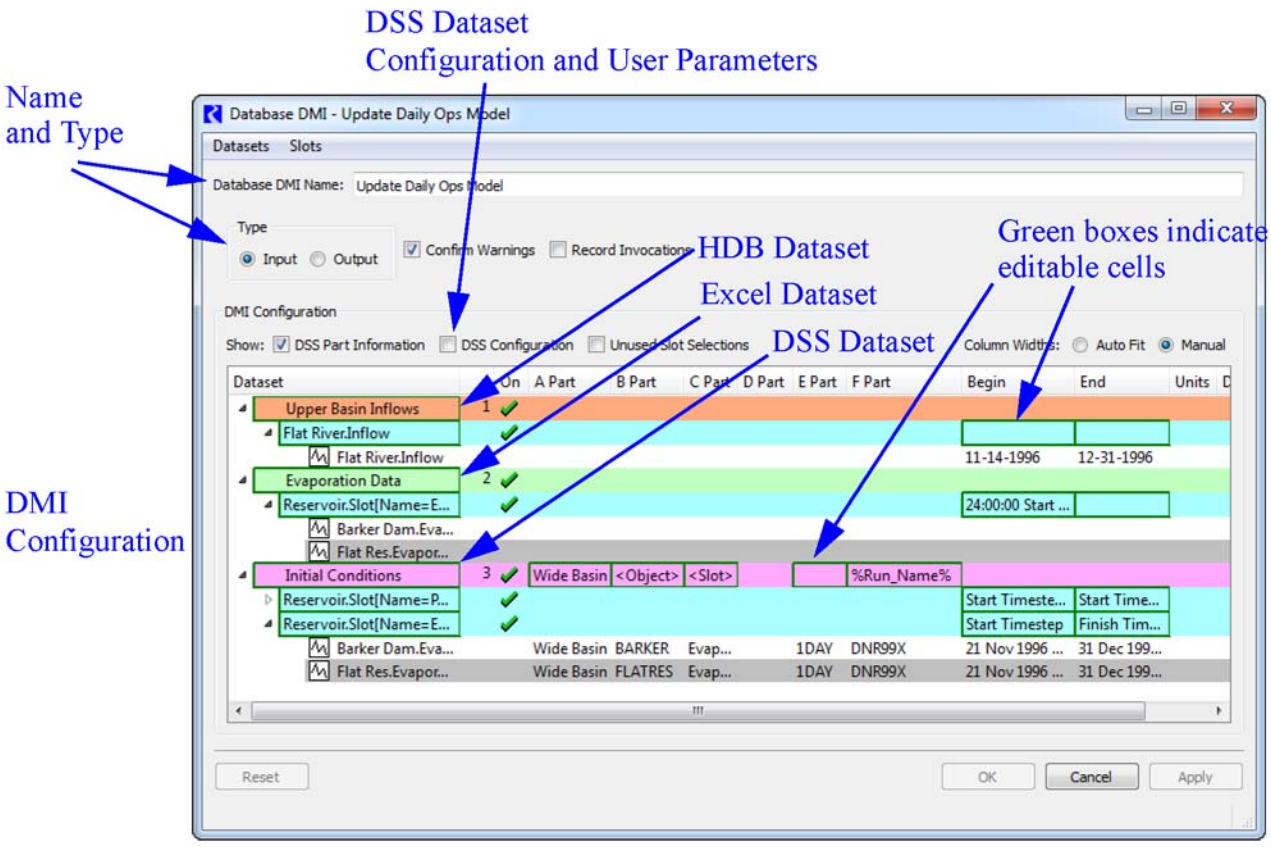
Conceptually the dialog has four sections from the top to the bottom, as shown in Figure 4.5 and described in the following sections:

- Database DMI Name and Type
- DMI Configuration
- DSS Configuration including DMI User Parameters (Only shown when a DSS dataset is present)

In this dialog, each type of dataset has its own color:

- HDB - Orange
- Excel - Green
- DSS - Pink

Figure 4.5



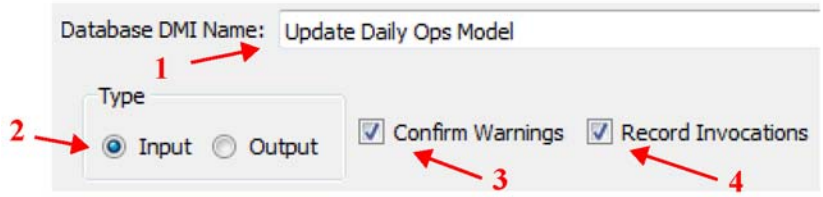
The cyan rows are slot selections. The green boxes are cells in which data entry is possible.

Name and Type

The Database DMI general configuration section (upper section) of the dialog allows you to do the following:

- Specify the **Database DMI Name** in the text field.
- Specify whether the Database DMI is an **Input** or **Output** DMI.
- Specify if you wish to confirm warnings before the Database DMI continues using the toggle.
- Specify whether he DMI manager should record invocations of the DMI. The **Record Invocations** check box allows RiverWare to maintain information which chronicles input DMI invocations. With this information, values set by DMIs can be cleared on a per-invocation basis. If a user wishes to clear out all values that are input by a DMI, this box should be checked. If a user knows values set by a particular DMI won't be cleared then it is not necessary to maintain information about the DMI invocations and this box should remain unchecked. (The information occupies memory and could, over time, degrade performance.) The checkbox is only enabled for

input DMIs and the default is unchecked. See "DMI Invocation Manager Dialog" on page 18 for details on clearing values set by an input DMI.



DMI Configuration

The DMI configuration section (middle section) displays the configuration of the DMI. In this dialog, each type of dataset has its own color:

- HDB - Orange
- Excel - Green
- DSS - Pink

The light blue (cyan) rows are slot selections. The green boxes are cells in which data entry is possible. This portion of the dialog allows you to:

- Create Datasets using the Dataset menu.
- Remove Datasets using right-click context menus.
- Create and define slots selections and time intervals.
- For DMIs with DSS datasets, see and edit the part information for each slot.
- Hide the part information using the **DSS Part Information** toggle.
- Show or hide Unused Slot Selections; see "Unused Slot Selections" on page 69.
- Select either **Auto Fit** or **Manual Column Widths**.

  Note: The first column in Auto Fit mode, only grows to a fixed maximum width.
- In this dialog, tool tips and the status bar (the lower left) are used to show the contents of cells, indicate directions, or additional information.

The information is presented in a tree-view, with each level representing a more detailed specification, as detailed in Table 4.1.

Table 4.1

Name	What it is	Tree Level	Items shown
Dataset	Entered by user	1: Highest	Name, type and generic Part information. The Name and Part information is editable. All Datasets including HDB and Excel Datasets, are listed. HDB and Excel datasets can be shown in this dialog but cannot be edited.
Slot Selection	Slot Selection Name as shown (or modified) in the Selector dialog. It can included wildcards, e.g. StorageReservoir.Outflow	2: Middle	The begin and end timestep for this selection are shown and can be edited.
Slots in Selection	Slots that were selected by the Slot Selection shown, including resolution of wildcards	3: Lowest	Fully specified information as it will be passed to the database. For DSS, this includes Part information, Begin and End Timestep, and units. No editing is allowed

The following sections describe the editable parts of each of the levels of the treeview.

Dataset

The following actions can be done for a Slot Selection row:

- Name DSS Datasets. Double-click to edit the name.
- Open HDB and Excel datasets. Double-click HDB or Excel datasets to edit.
- Disable a Dataset for debugging; see "Enabling and Disabling for Debugging" on page 68.
- Specify the part information. There are many options for specifying the part information. The only restriction is that Part D is not editable and the default for Part E is the timestep size. The part information can be edited by entering text or by selecting text from a right-click context menu. In addition to typing text, you can use the two following items to generalize DSS DMIs:
  - **Workspace Entities**.You can use generic Workspace Entities for wildcarding any part (except Part D which is not editable). Select the desired entity from the right-click context menu. Table 4.2 lists the options.

Table 4.2

Workspace Entity	Description
Slot	The slot name.
Parent	The item that contains the slot. For a simulation slot, this is the object (e.g. a reservoir). For an accounting slot, this is the account (e.g. a storage account).
SimObj	The object that contains the item. For both simulation and accounting slots, this is the containing object (e.g. a reservoir)
Account	The name of the account containing the object. (Valid when accounting is enabled)
Supply	The name of the supply. For example a supply named, ResAStorageToReachBFish.Supply would result in ResAStorageToReachBFish (Valid when accounting is enabled)
Exchange	The name of the exchange. (Valid when accounting is enabled)
Payback	The name of the payback. (Valid when accounting is enabled)

Table 4.2

Workspace Entity	Description
UpSimObj	Not implemented yet. For a supply, this evaluates to the object that contains the upstream account.
UpAccount	Not implemented yet. For a supply, this evaluates to the account name of the upstream account.
DownSimObj	Not implemented yet. For a supply, this evaluates to the object that contains the downstream account.
DownAccount	Not implemented yet. For a supply, this evaluates to the account name of the downstream account.

Note: In versions prior to 6.1, Part B was by default the object name and Part C was by default the slot name. Now these need to be explicitly defined for new datasets, existing datasets will be given the <parent> and <slot> entities for parts b and c, respectively.

– User Parameters. You can use parameters in the DSS parts (see "DSS DMI User Parameters" on page 71, for details). To use the parameters in the DSS part information, specify the value with the percent sign:

```
%Parameter%
```

Slot Selection

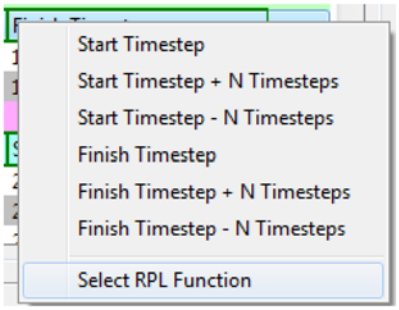
The following actions can be done for a Slot Selection row:

- Select slots by selecting the name of the slot selection. Typically, this is NONE before the slots have been selected. Use the Selector to choose the desired slots. Either create a new selection or use an existing Slot Set (see "Slot Sets" in *User Interface*). When creating a selection, wildcards are very useful to symbolically specify the slots. For example, all Reservoir Inflows could be specified using a wildcard on reservoirs and on the slot name, Reservoir.Slot[Name=Inflow]. See "Creating a Name Map of All Reservoir.Inflow Slots" in *User Interface* for a step by step example of this selection.

  Note: The Slot Selections are maintained as references to the specified slots. So if you delete the objects/slots to which the selection refers, the references could be lost, particularly if the slots are manually specified, that is, not using wildcards.The references are resolved when the Database DMI is opened. So if you want to export and re-import the same named data object (maybe from a different model) without breaking the DMI references, close all DMI dialogs, delete the object, re-import the new object and then open the DMI dialogs. The references will resolve and should point to the new object/slots. If you do this out of order, the references may be lost.

- Disable a Slot Selection for debugging; see d "Enabling and Disabling for Debugging" on page 68.
- Specify the time interval for each slot selection. The begin and end dates are edited in place; they can be edited by entering text or by selected text from a right-click context menu. In both cases the text is a symbolic date/time. If not entered, the default time range is the slot's range. In the Begin and End columns, the user specifies the timesteps at which to begin/end the Input or Output data. The user can either select and manually type in the date, or right-click and a select from a list of commonly used reference datetimes as shown in the following

figure. If selecting one of the reference datetimes, the user may enter the number of Timesteps in place of the letter N as necessary.

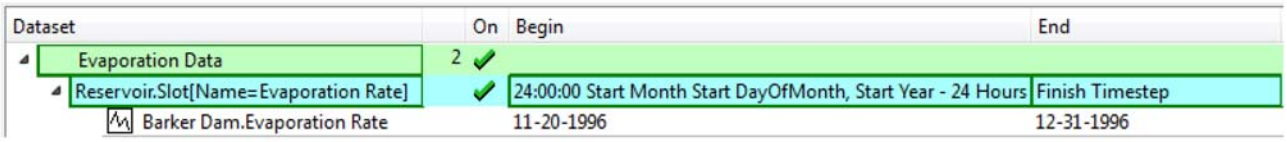


The datetime uses the same syntax as the RiverWare Policy Language (RPL) datetimes. For example, as shown in Figure 4.6, the user could enter either of the following as a valid datetime:

– `1:00 April max DayOfMonth, 2006`

– `24:00:00 Start Month Start DayOfMonth, Start Year - 24 Hours`

Figure 4.6

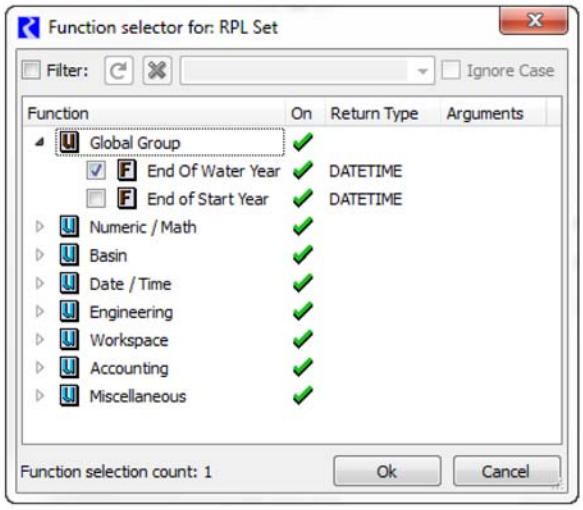


See "DATETIME" in *RiverWare Policy Language (RPL)* for details on datetimes. Basically, any fully specified datetime can be used

Note: No @ or quotes " " are necessary when specifying the datetimes in the DMI slots dialog.

To allow additional flexibility specifying datetimes, the user can choose the **Select Function From Global Function Set** option and then choose a RPL function from an opened Global Function Set; see "Global RPL Functions" in *RiverWare Policy Language (RPL)* for details. A RPL Object selector opens and displays a tree

view of all the global utility groups and predefined function groups; see "Example: Creating a New RBS Rule-set" in *RiverWare Policy Language (RPL)* for details.



The user then selects a single function by selecting the box to add a check mark. This function MUST:

– Not have any arguments

– Return a fully specified DATETIME variable

If either is not true, an error will be posted when the selection is applied or the DMI is run. After selecting **Ok**, The function name is displayed in the Begin / End column with "( )" appended. Alternatively, a user may manually type in the function name, but must include the "( )".

Slots in Selection

The following actions can be done for Slot Selection rows:

- See the slots that were selected using wildcard.
- See the resolved time interval for each slot, as it will be passed to DSS.
- See the units and scale as they will be passed to DSS.

- For DSS parts, mouse over the slot's part columns to see the information that will be passed to DSS:

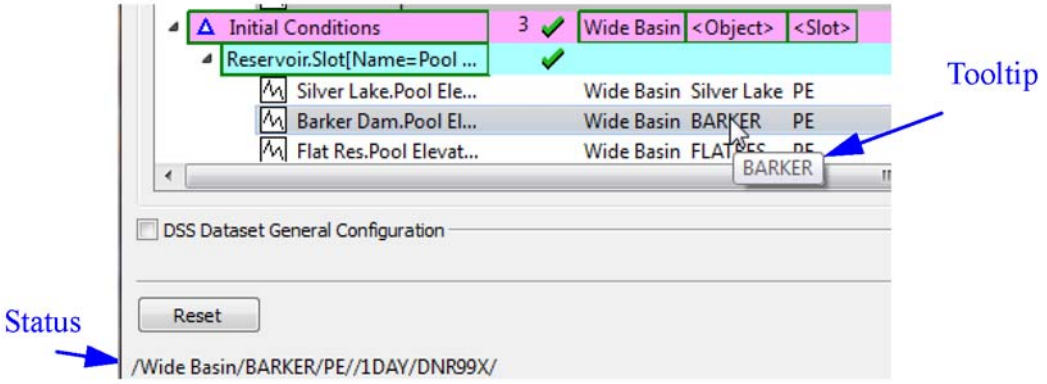


Figure 4.7 shows a fully defined DSS configuration.

Figure 4.7

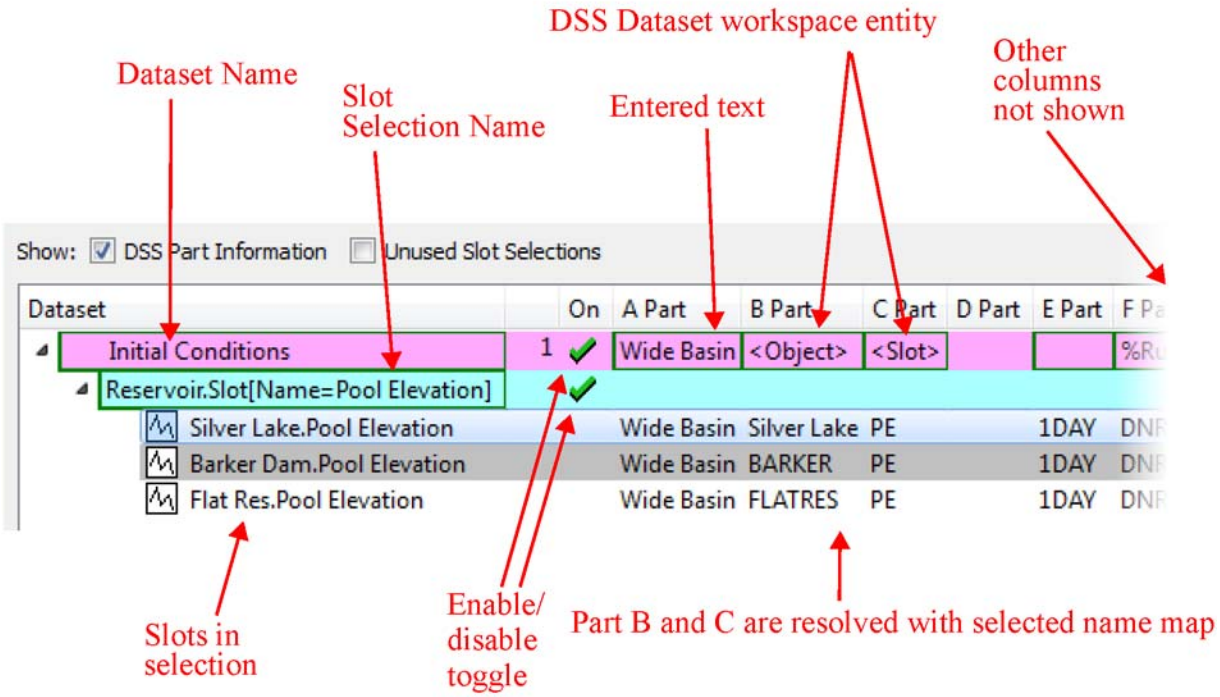
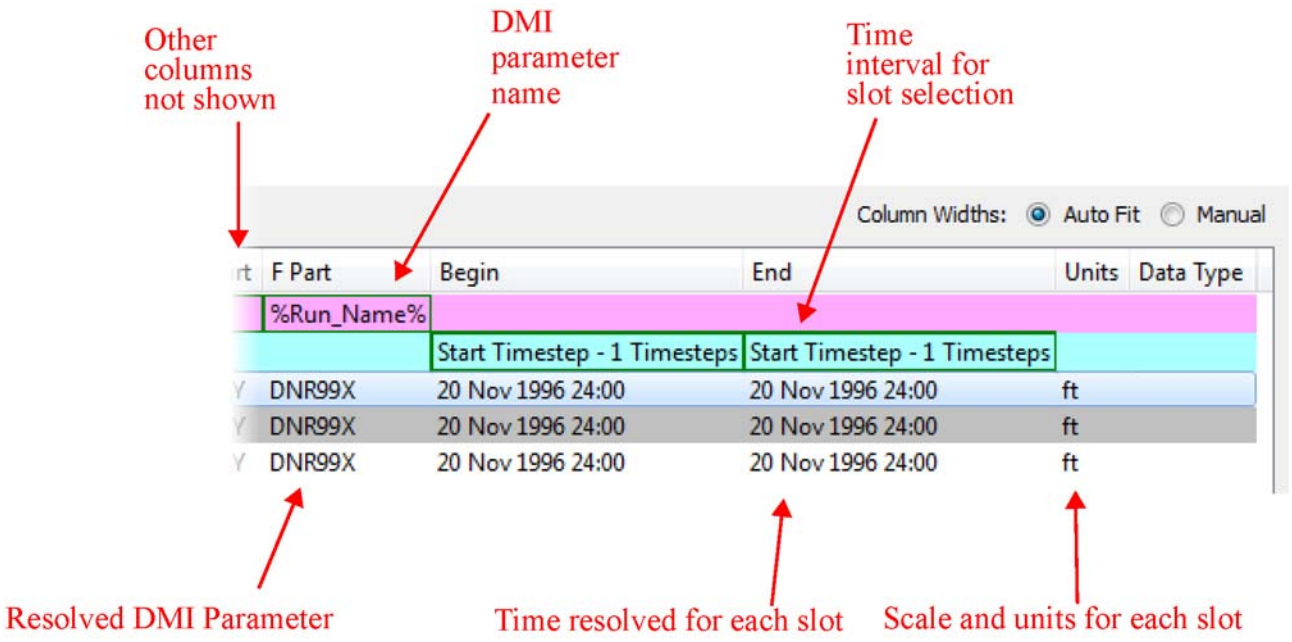


Figure 4.8 shows the right portion of the dialog.

Figure 4.8



Enabling and Disabling for Debugging

Datasets and slot selections can be disabled for debugging purposes. On the left part of each Dataset or Slot selection row, there is a green check if it is enabled. A red X indicates it is disabled. The purpose of this feature is to allow you to debug a DMI by disabling datasets or slot selections and then running the DMI to test other aspects of the DMI. This behaves as follows:

- Turning off a dataset causes its slot selections to be turned off.
- Turning off all slot selections causes the dataset to be shown as off.
- The dataset is shown as a tristate (on/green, off/red, partially on/orange) based on the state of its slot selections.

The on/off state is preserved in the model file when saved.

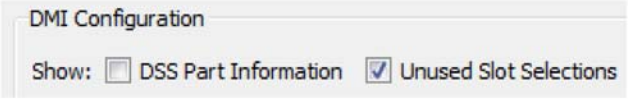
Use of Name Maps

If you specify a Name Map when configuring the Dataset, then the following will occur:

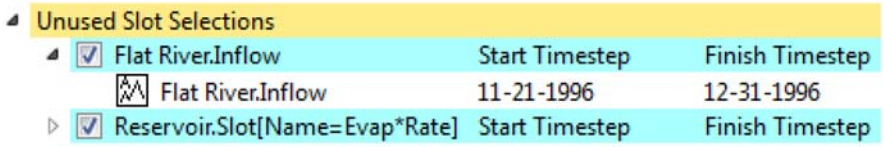
- If the RiverWare Object is in the Name Map then its mapped name will be used for <Object> or <Parent> entities; otherwise, its RiverWare name will be used.
- If the Slot is in the Name Map then its mapped name will be used for <Slot> entities otherwise, its RiverWare name will be used.
- If a Table Slot column is in the Name Map then its mapped name will be used for Part C; otherwise, the Table Slot's RiverWare name will be used.

Unused Slot Selections

It is possible to have slot selections that are not used by any of the datasets. To show these selections, select **Unused Slot Selections**.



The selections are then shown as a golden colored row titled Unused Slot Selections at the Dataset level. You can view and change the selections (and the slots selected). From right-click context menus, you can delete the selection. You can also drag and drop the slot selection onto a dataset to associate that slot selection with a dataset.



DSS Dataset Configuration

For DSS datasets, an additional panel allows you to fully edit the dataset.

> **Note:** This is not available for Excel or HDB. For those datasets, you can double-click the dataset and it will open the dataset editor.

When there is a DSS dataset in the DMI, the DSS checkboxes are displayed. Select the **DSS Configuration** check box to show general configuration settings.

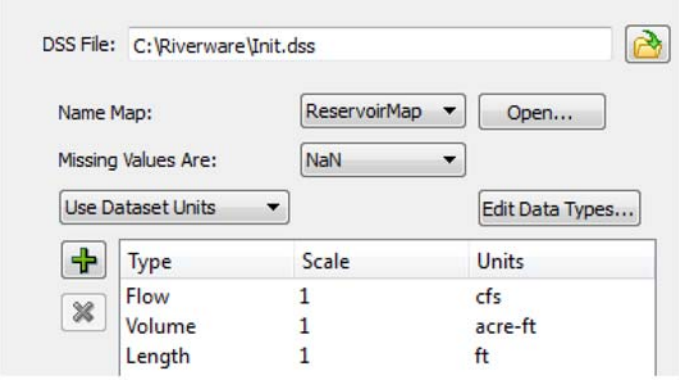


This shows a view similar to the dataset. When you have a DSS dataset selected, the general configuration becomes enabled and allows you to:

- Specify the DSS file for a DSS Dataset.
- Specify the name map using the pull down menu and/or **Open** the name map manager.
- Specify the missing value behavior for the selected DSS Dataset.
- Specify the units for the selected DSS Dataset.
- Configure Data Types by selecting the **Edit Data Types** button.

See "Datasets Overview" on page 82 for descriptions of these fields.

Note: Each of these specifications only works on the **selected dataset**. In other words, you must highlight a dataset row from the middle portion of the dialog before it allows you to change units, scale, name map to use, or how missing values are treated.

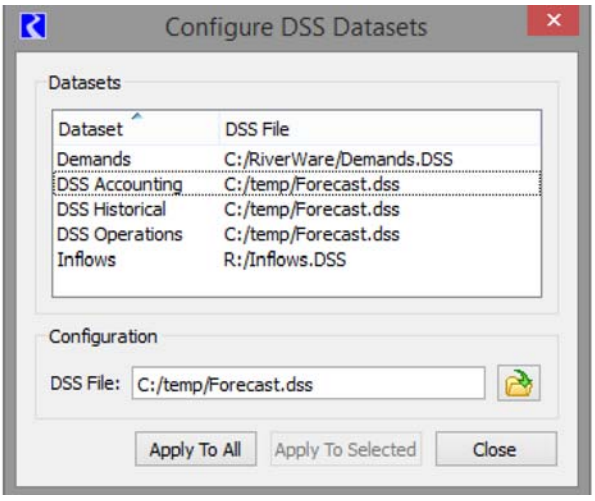


Setting Multiple DSS Dataset Files

You can configure the DSS file path for multiple DSS datasets through the Configure DSS Datasets dialog. The dialog is opened by

- Right selecting a DSS dataset in the Database DMI editor dialog and selecting the **Set Multiple DSS Dataset Files** menu item, or
- Selecting a DSS dataset in the database DMI and then using the **Datasets**, then **Set Multiple DSS Dataset Files** menu.

The Configure DSS Datasets dialog appears.

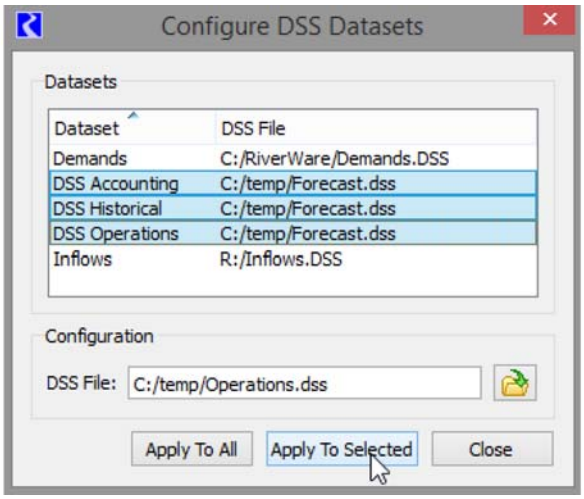


The top portion of the dialog shows all of the DSS Datasets in the model and their DSS File in sortable columns. The bottom portion of the dialog shows the file path for the dataset that was selected when the dialog opened.

Specify an alternative DSS file if desired. Then:

- Select one or more datasets to modify and choose **Apply to Selected** as shown below.

- Select **Apply to All** to change every dataset to the new file path.



If a DSS dataset has been modified in another dialog, the DSS file cannot be set. You are notified with the following message: “The following datasets cannot be modified because they are being modified in another dialog”. Apply the changes to that dialog and you can now use the Configure DSS Datasets dialog again.

DSS DMI User Parameters

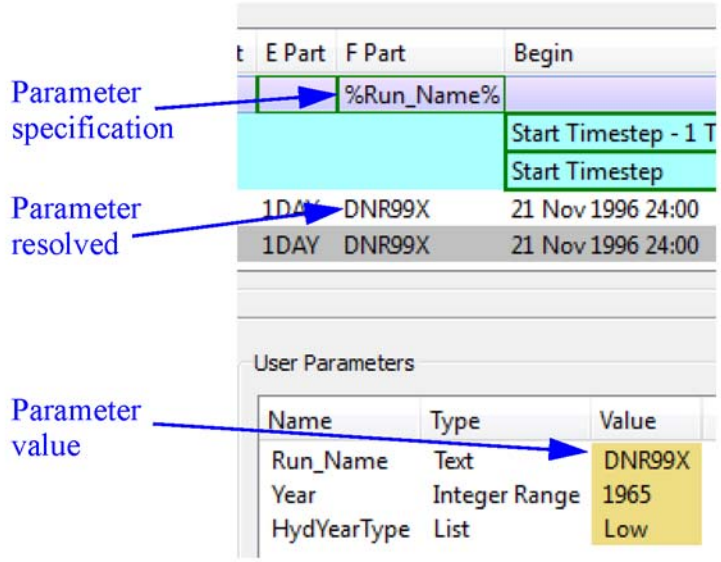
For DSS, you can specify values to be used for the user parameters. The parameters must be previously configured using the Parameter Dialog; see “DMI Parameter Dialog” on page 11 for details.

Edit a parameter value by double-clicking the Value cell to activate a control appropriate for the parameter’s type. To use the parameters in the DSS part information, specify the value with the percent sign; use the following syntax:

```
%Parameter%
```

Thus if you have a parameter named Run_Name with a text value and you wanted to put the value in part F, you would double-click part F on the desired Dataset row and type `%Run_Name%`. In the lower portion, double-click the value column and specify the integer to use. The resolved information is shown for each slot that is a part of the dataset. Figure 4.9 shows this example.

Figure 4.9



Examples

Now that we have described the overall approach for creating a Database DMI, we will present a few example use cases of creating various types of Database DMIs.

Example 1: DSS DMI Using the Database DMI Editor

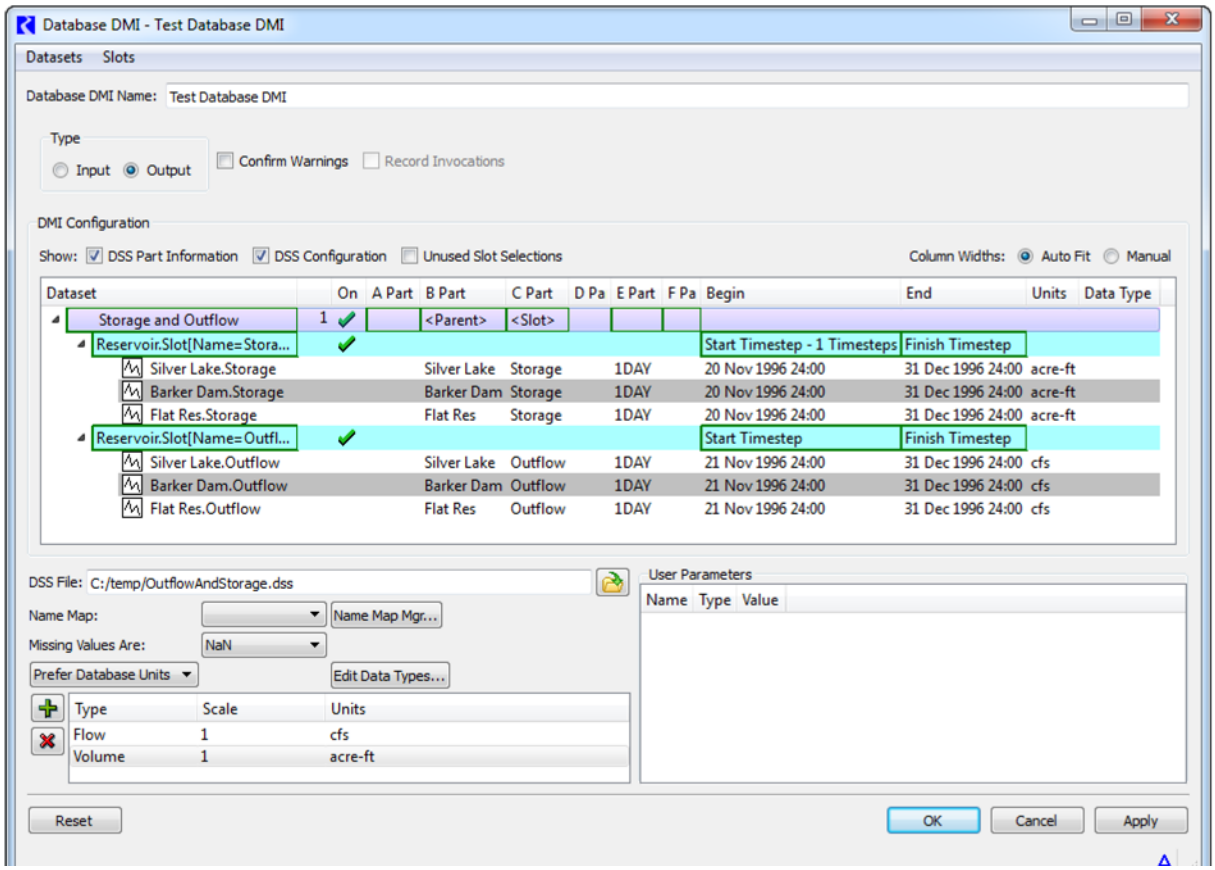
This example shows how to create a Database DMI to/from DSS using the Database DMI Editor. We assume there are no other Datasets or Name Maps already defined. You would take the following steps:

1. From the workspace, open the DMI Manager using the **DMI**, then **DMI Manager.**
2. Create a new Database DMI using the **DMI**, then **New Database DMI** menu.
3. Open the Database DMI Editor using the **Edit**, then **Edit** menu.
4. In the upper portion of the dialog, enter a name, whether it is an Input or Output DMI, whether to confirm warnings, and whether to record invocations; see "Name and Type" on page 61. Let's assume we are creating an Output DMI. Leave everything else unchecked.
5. First, add a dataset using the **Datasets**, then **New DSS Dataset** menu.
6. Expand the Dataset column and double-click the Dataset0 name and edit it.
7. Next, we will select the slots that we wish to output, say all Reservoir.Storages and Reservoir.Outflows. In the middle portion of the window, right-click the dataset you added, then use the **New Slot Selection**. A row is added to the tree view below the dataset.
8. Double-click the word NONE to open the slot selector. Use the selector to choose Reservoir.Storage. Add another slot selection and select Reservoir.Outflow. See "Creating a Name Map of All Reservoir.Inflow Slots" in *User Interface* for an example of a similar selection.

9. For the Reservoir.Outflow row, specify the timestep by right-clicking once in the **Begin** field. Choose “Start timestep”. In the End column right-click once and choose “Finish timestep”. For the storage row, repeat and choose “Start Timestep - N” to “Finish Timestep”. Double-click the “N” in the Begin column and type “1” to specify 1 timestep.
10. On the dataset row, right-click in the B Part and select Workspace Entity. Choose <Parent>.
11. On the dataset row, right-click in the C Part and select Workspace Entity. Choose <Slot>.
12. Select the treeview symbol next to the Reservoir.Outflow row to see the exact slots you are going to output and all the part information, timestep, and units. We could change any of the part information by selecting the appropriate column on the Dataset row. For example, we could enter OBS for observed in part F by selecting that field on the dataset row and typing it.
13. Now configure the dataset. Select the dataset row. In the middle of the dialog, select the **DSS Configuration** toggle.
14. Specify the name of a **DSS File** in which to send the output either by typing it or using the selector. On an Input DMI, this file must already exist. On the Output DMI, it will be created if it does not exist.
15. If we have already configured a **Name Map**, we could select that we wanted to use it in the middle portion of the dialog; see “DSS Dataset Configuration” on page 69. A name map is created outside of this dialog; see “Name Mapping” on page 117. Because this is an Output DMI, we do not necessarily have to have a Name Map.
16. Now let’s specify the units to write. In the middle section, select the “+” button. Repeat to add two rows. See “DSS Dataset Configuration” on page 69 for details.
17. Double-click the word NONE under the type column to activate the pull-down menu. Choose **Volume**. On the Units column repeat and choose the desired unit, say **Acre-feet**. On the second row, Choose **Flow** and **cfs** as the units.
18. The Database DMI is now fully configured. Select **OK** to close the dialog and apply the changes. Invoke the DMI from the DMI manager using the **DMI**, then **Invoke** menu.

Figure 4.10 shows the configuration for this example.

Figure 4.10



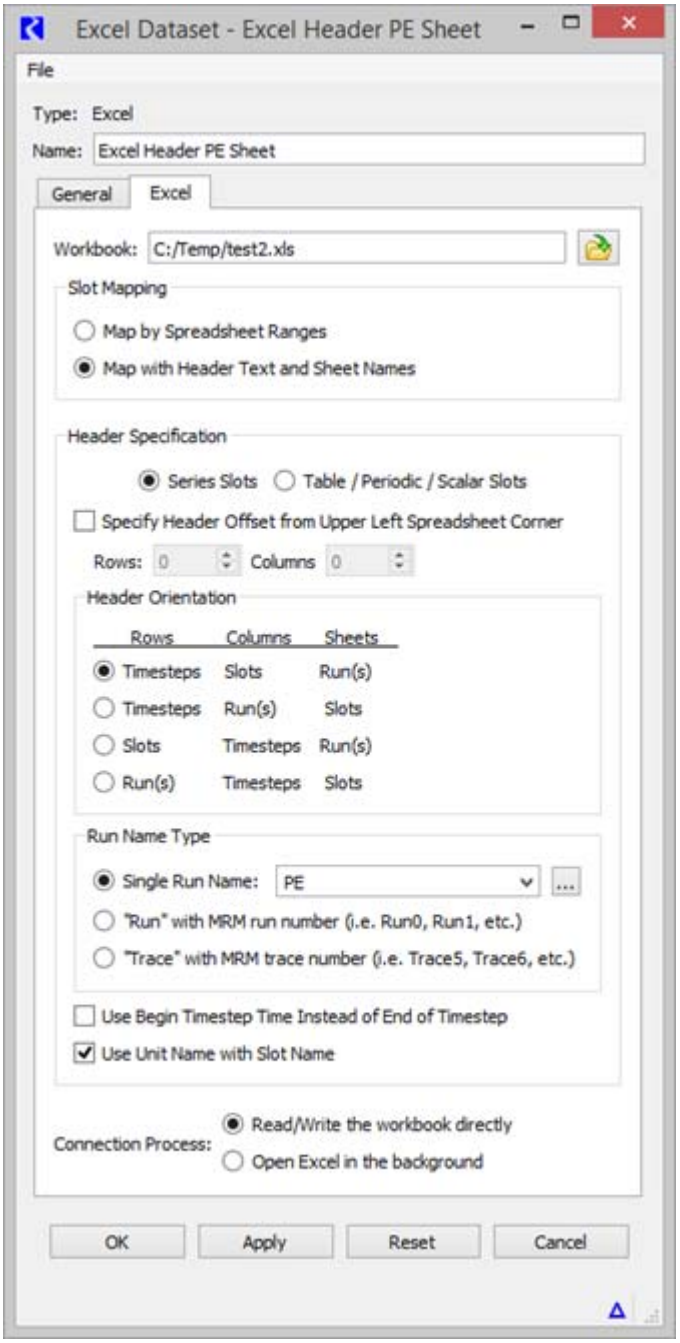
Example 2: Excel DMI to Export Data Using Headers

This example shows how to create a Database DMI to export data to Excel using the Header approach. We assume there are no other Datasets or Name Maps already defined. You would take the following steps:

1. From the workspace, open the DMI Manager using the **DMI**, then **DMI Manager.**
2. Choose **Utilities**, then **Datasets**
3. Choose **Dataset**, then **New Excel Dataset**
4. Double-click the newly created dataset to open it.
5. You will create an output DMI so specify a name in the **Name:** field. Say “Excel Header PE Sheet”.
6. Select the **Excel** tab and type in or navigate to a worksheet. Because it is an output DMI and you don’t have an existing sheet, you will have to type in a file name in the selector.
7. Now you will need to choose which approach you wish to use: Ranges or Headers. If you choose Ranges, you will need to create Name Maps. See “Slot Mapping—Spreadsheet Ranges Approach” on page 105 for similar samples, based on the configuration. Let’s use the header approach (see “Slot Mapping—Header Text and Sheet Names Approach” on page 109). Select the **Map with Header Text and Sheet Names** option.
8. You are exporting series slot data, so select the Series Slots option.

9. Specify the Header orientation and specify a sheet name, for example PE. In this sample, use the Timestep, Slots, Runs option.
10. Optionally, specify to **Use Unit Name with the Slot Name**. Figure 4.11 shows the setup for the Excel tab. With this setup, you do not have to use Name Maps, but you could. Thus, we have a fully configured dataset.

Figure 4.11



11. Select **OK** to accept the changes and close the dialog

12. Back in the DMI Manager, create a new Database DMI using the **DMI**, then **New Database DMI** menu.
13. Open the Database DMI Editor using the **Edit**, then **Edit** menu.
14. In the upper portion of the dialog, enter a name, that it will be an Output DMI, and whether to confirm warnings. See "Name and Type" on page 61 for details.
15. Now add the dataset you configured earlier. Choose **Datasets**, then **Add Existing** to open the Dataset Selector dialog.
16. Choose the "Excel Header PE Sheet" dataset and select **OK**.
17. Next, we will select the slots that we wish to output, say all Reservoir Pool Elevations. With the Excel Header PE Sheet row highlighted, use the **Datasets**, then **New Slot Selection** menu.
18. Double-click the word NONE to open the slot selector. Use the selector to choose all Reservoir.Pool Elevations. See "Creating a Name Map of All Reservoir.Inflow Slots" in *User Interface* for an example of a similar selection.
19. For the Reservoir.Pool Elevation row, specify the begin timestep by right-clicking once in the **Begin** field. Choose "Start timestep". In the **End** column right-click once and choose "Finish timestep".

Figure 4.12 shows this setup.

This database DMI is now fully configured. You can execute it from the DMI manager.

Figure 4.12

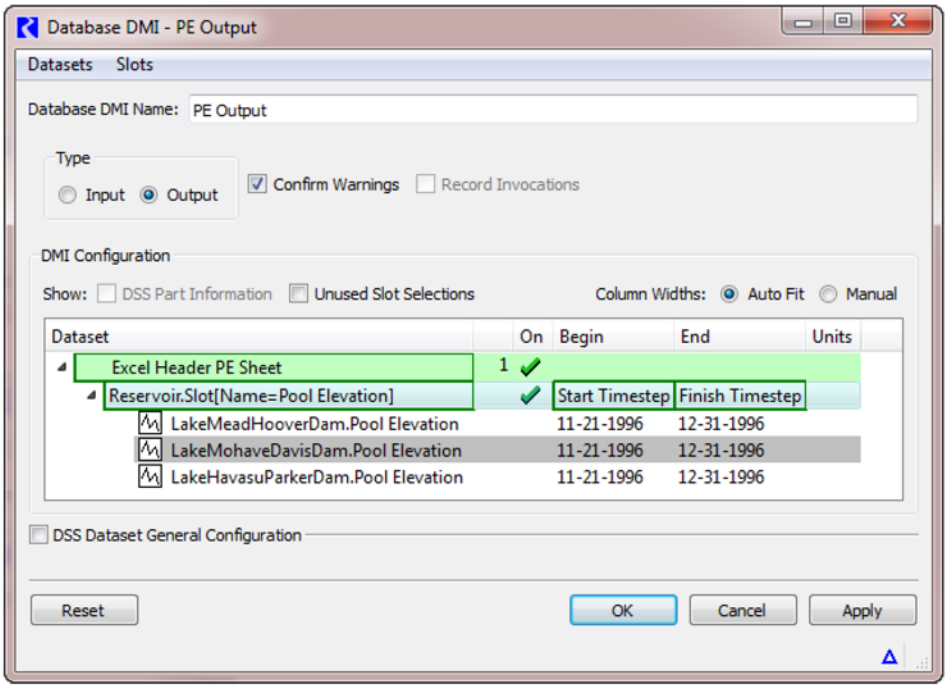
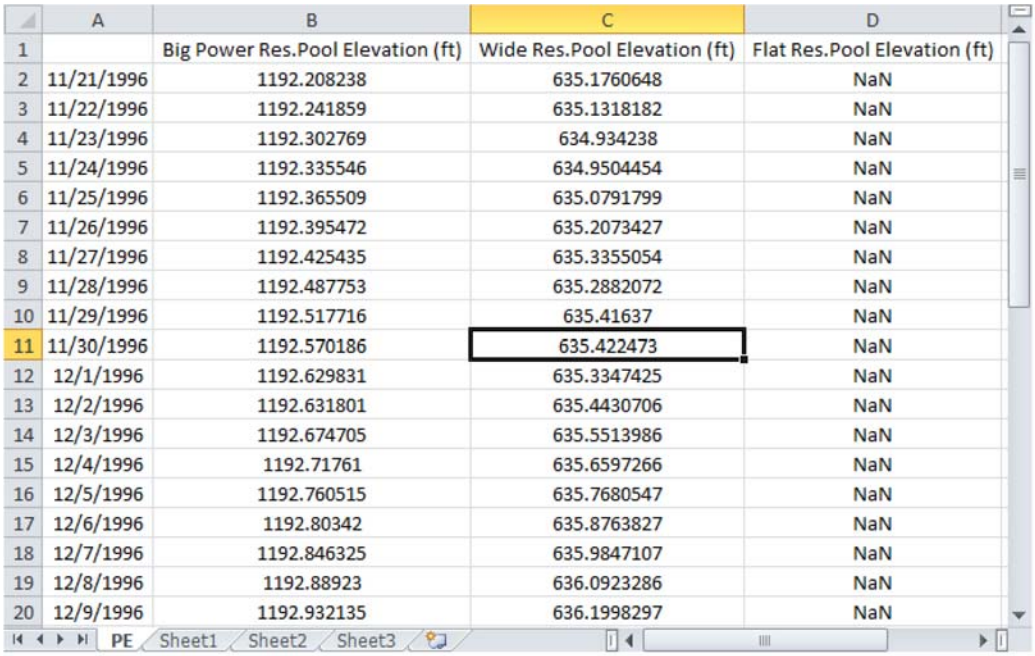


Figure 4.13 shows the resulting spreadsheet for our three-reservoir model.

Figure 4.13



	A	B	C	D
1		Big Power Res.Pool Elevation (ft)	Wide Res.Pool Elevation (ft)	Flat Res.Pool Elevation (ft)
2	11/21/1996	1192.208238	635.1760648	NaN
3	11/22/1996	1192.241859	635.1318182	NaN
4	11/23/1996	1192.302769	634.934238	NaN
5	11/24/1996	1192.335546	634.9504454	NaN
6	11/25/1996	1192.365509	635.0791799	NaN
7	11/26/1996	1192.395472	635.2073427	NaN
8	11/27/1996	1192.425435	635.3355054	NaN
9	11/28/1996	1192.487753	635.2882072	NaN
10	11/29/1996	1192.517716	635.41637	NaN
11	11/30/1996	1192.570186	635.422473	NaN
12	12/1/1996	1192.629831	635.3347425	NaN
13	12/2/1996	1192.631801	635.4430706	NaN
14	12/3/1996	1192.674705	635.5513986	NaN
15	12/4/1996	1192.71761	635.6597266	NaN
16	12/5/1996	1192.760515	635.7680547	NaN
17	12/6/1996	1192.80342	635.8763827	NaN
18	12/7/1996	1192.846325	635.9847107	NaN
19	12/8/1996	1192.88923	636.0923286	NaN
20	12/9/1996	1192.932135	636.1998297	NaN

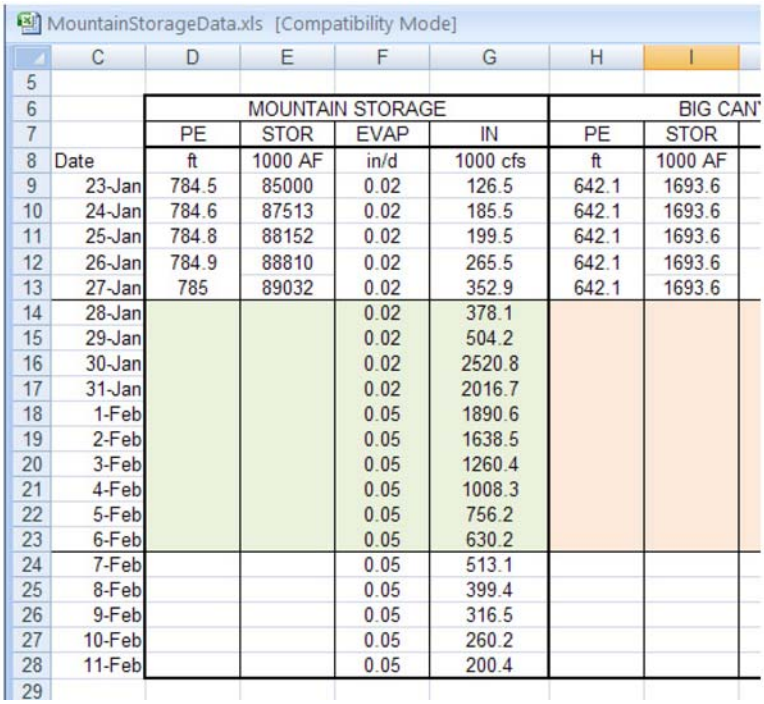
PE Sheet1 Sheet2 Sheet3

Example 3: Excel DMI to Import Data Using Ranges

This example shows how to create a Database DMI to import data from Excel using the Ranges approach. We assume there are no other Datasets or Name Maps already defined. You would take the following steps:

1. Figure 4.14 is screenshot of the desired data in Excel. It is not in the format necessary for the Headers approach. So we will use the Ranges approach. We would like to import the Mountain Storage IN column to the slot Mountain Storage.Inflow for just the run range (Jan 28-Feb 6th). That is, we want to import the range G14:G23.

Figure 4.14



MountainStorageData.xls [Compatibility Mode]

	C	D	E	F	G	H	I
5							
6		MOUNTAIN STORAGE				BIG CAN'	
7		PE	STOR	EVAP	IN	PE	STOR
8	Date	ft	1000 AF	in/d	1000 cfs	ft	1000 AF
9	23-Jan	784.5	85000	0.02	126.5	642.1	1693.6
10	24-Jan	784.6	87513	0.02	185.5	642.1	1693.6
11	25-Jan	784.8	88152	0.02	199.5	642.1	1693.6
12	26-Jan	784.9	88810	0.02	265.5	642.1	1693.6
13	27-Jan	785	89032	0.02	352.9	642.1	1693.6
14	28-Jan			0.02	378.1		
15	29-Jan			0.02	504.2		
16	30-Jan			0.02	2520.8		
17	31-Jan			0.02	2016.7		
18	1-Feb			0.05	1890.6		
19	2-Feb			0.05	1638.5		
20	3-Feb			0.05	1260.4		
21	4-Feb			0.05	1008.3		
22	5-Feb			0.05	756.2		
23	6-Feb			0.05	630.2		
24	7-Feb			0.05	513.1		
25	8-Feb			0.05	399.4		
26	9-Feb			0.05	316.5		
27	10-Feb			0.05	260.2		
28	11-Feb			0.05	200.4		
29							

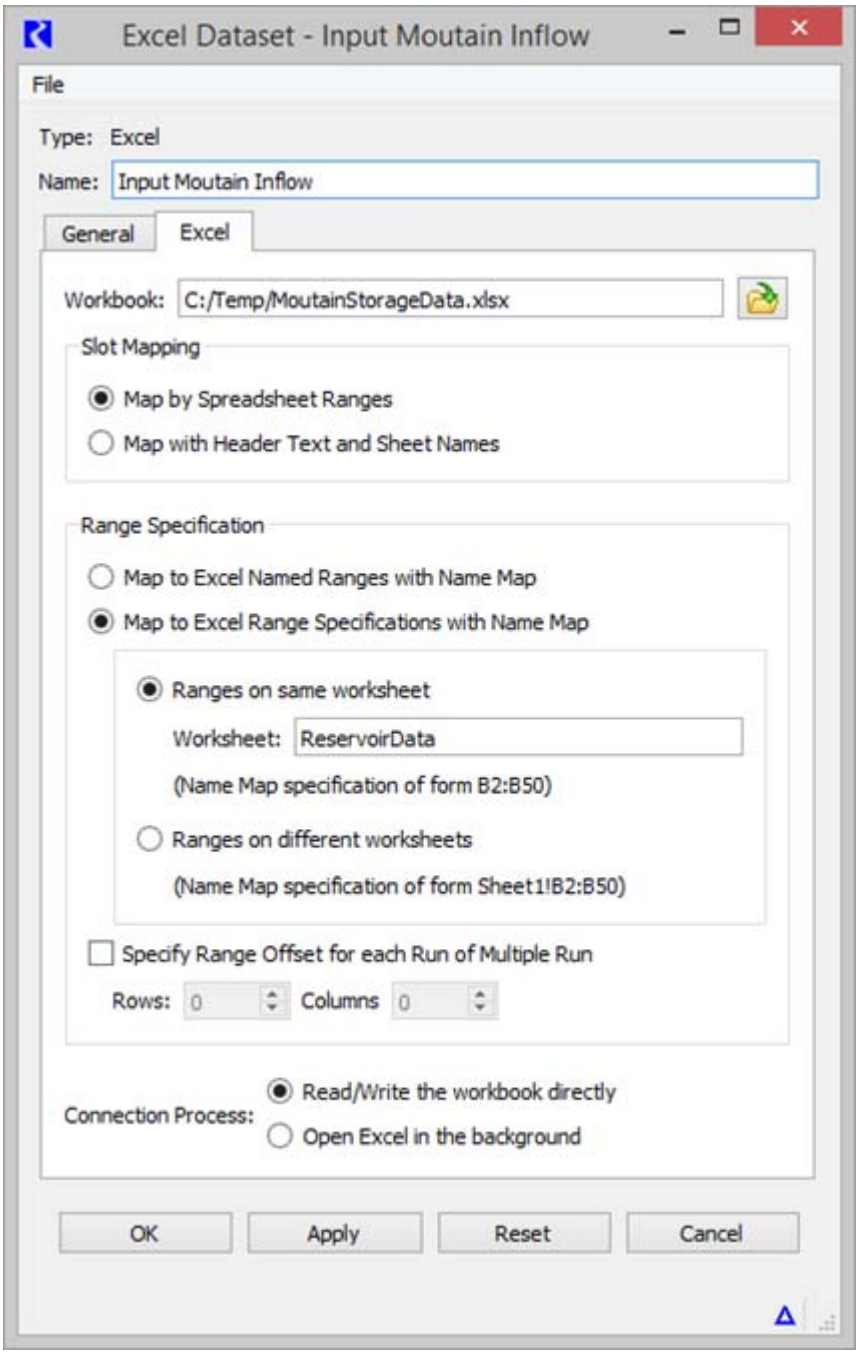
2. From the workspace, open the DMI Manager using the **DMI**, then **DMI Manager.**
3. From the DMI Manager choose **Utilities**, then **Name Maps**.
4. Choose **Name Map**, then **New** or select the + button.
5. Double-click the newly created Name Map.
6. Change the name to "`Mountain In`"
7. Select **Name Map**, then **New** to add a row to the list.
8. Click-pause-click on the word **NONE** and choose **Select Slot**.
9. Using the selector, choose the reservoir slot **Mountain Storage.Inflow**. Select **OK**.
10. In the Name Map, in the **Name** column type "`G14:G23`" (without quotes).



11. Select **OK** in the Name Map and **OK/Cancel** in the Name Map Manager to close.
12. In the DMI Manager, choose **Utilities**, then **Dataset.**
13. Choose **Dataset**, then **New Excel Dataset**.
14. Double-click the newly created dataset to open it.

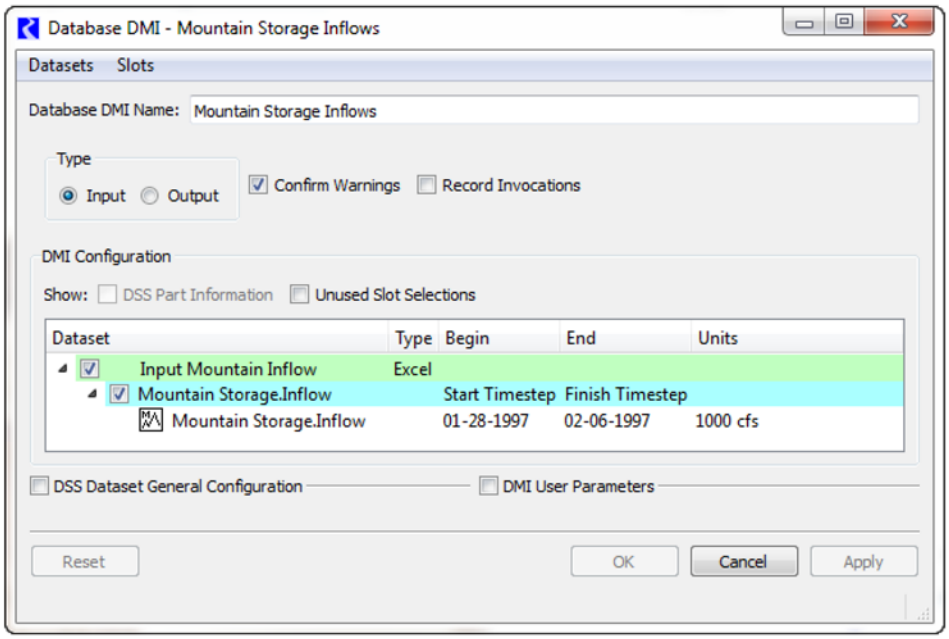
15. You will create an input DMI so specify a name in the **Name:** field: "`Input Mountain Inflow`".
16. In the Name Map section, choose the Name Map you created "Mountain In" from the pull down menu.
17. Select **Units**, then **New**
18. Under the **Type** column, click-pause-click and choose **Flow** from the pull down menu.
19. Change the **Scale** to 1000 and the **Units** to cfs.
20. Select **Apply**.
21. Select the **Excel** tab and type in or navigate to the worksheet where the data resides. In the sample, it is in `C:/Temp/MountainStorageData.xls`.
22. Select the **Map by Spreadsheet Ranges** button.
23. Choose the **Map to Excel Range Specification with Name Map**.
24. The data is all on the same worksheet and we didn't enter it in the Name Map, so you will need to enter it here. In the **Worksheet** field, enter the name of the sheet, "ReservoirData".
25. The dataset is fully configured, as shown in Figure 4.15. Select **OK** to apply and close the dialog.

Figure 4.15



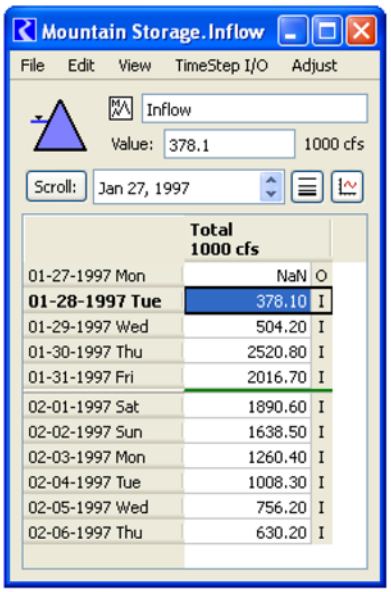
26. Back in the DMI Manager, create a new Database DMI using the **DMI**, then **New Database DMI** menu.
27. Open the Database DMI Editor using the **Edit**, then **Edit** menu.
28. In the upper portion of the dialog, enter a name and that it will be an **Input DMI**. See "Name and Type" on page 61 for details.

29. Now add the dataset you configured earlier. Choose **Datasets**, then **Add Existing** to open the Dataset Selector dialog.
30. Choose the **Input Mountain Inflow** dataset and select **OK**.
31. We will select the slots that we wish to input. With the Input Mountain Inflow row highlighted, select **Datasets**, then **New Slot Selection** menu.
32. Double-click on the word **NONE** to open the slot selector. Use the selector to choose Mountain Storage.Inflow.
33. For the Mountain Storage.Inflow row, specify the timestep by right-clicking once in the **Begin** field. Choose "Start timestep". In the **End** column right-click once and choose "Finish timestep".



34. This database DMI is now fully configured. You can execute it from the DMI manager. Figure 4.16 shows the resulting slot.

Figure 4.16

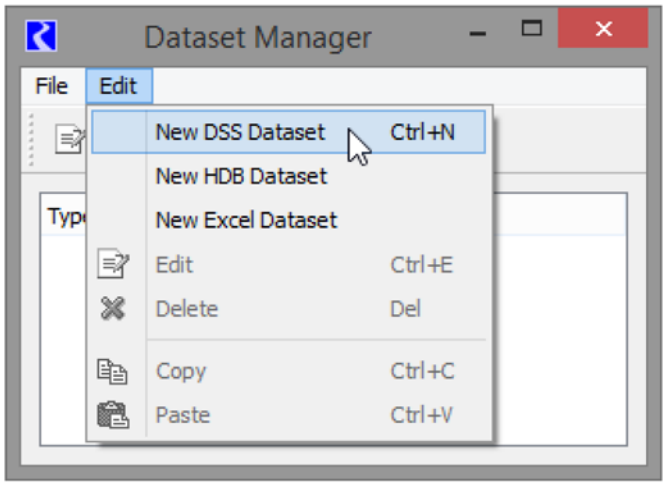


Datasets Overview

A dataset encapsulates all RiverWare knowledge about the database. In general, there are two types of knowledge contained in a dataset, configuration information and the database specification.

In RiverWare, datasets are defined and configured in the Dataset Manager as shown in Figure 4.17. The Dataset Manager is accessed from the **Utilities**, then **Dataset** menu on the DMI Manager. Selecting **Edit**, then **New DSS Dataset**, for example, adds a new DSS dataset.

Figure 4.17



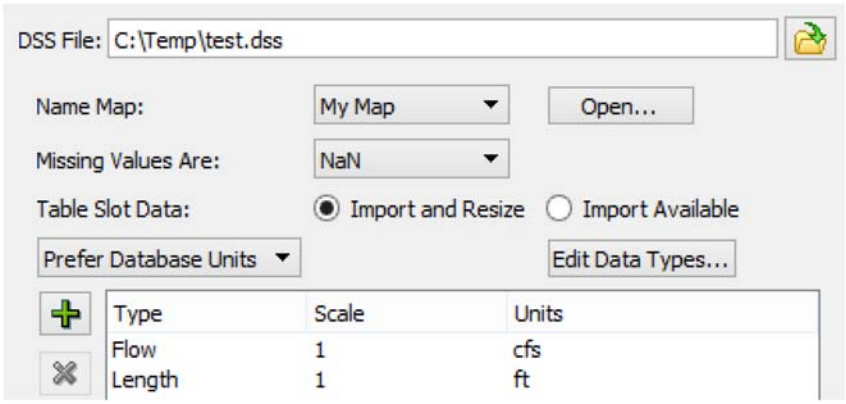
DSS Datasets

In general, all of the DSS dataset configuration can (and should) be performed from the Database DMI Editor. The DSS Dataset editor shows the results of configuration but the Database DMI has much more functionality. This section describes the functionality of the Database DMI Editor with respect to DSS datasets.

Note: A DSS dataset must be selected for this area to become active.

DSS File

Specify the DSS file either by typing in the file path or selecting it using the open file icon . If not already created, an export DMI will create the specified DSS file.



The DSS File name may contain user parameters as %ParamName% (the same as for the DSS part information). The DSS file name may contain multiple parameters but the parameters must be typed in; they are not available from a context menu. See "DSS DMI User Parameters" on page 71 for details.

The DSS file name text shows the user parameters while a tooltip shows the substituted values.The user parameters can be all or part of the DSS file name and can be used in conjunction with environment variables, for example:

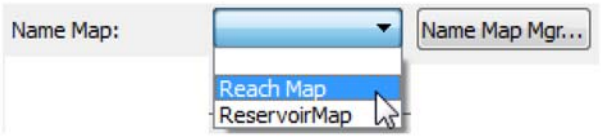
- A single parameter: `%DssFile%`
- A path and a parameter: `C:\model\data\%DssFile%`
- Multiple parameters: `%ModelDir%\data\%DssFile%` or `%DataDir%\%DssFile%.dss`
- Environment variable and a parameter: `$(DATA_DIR)\%DssFile%.dss`

Note: You can set the file path for multiple DSS Datasets in one action using the Configure DSS Datasets dialog; see "Setting Multiple DSS Dataset Files" on page 70 for details.

Name Map

Optionally, select the desired Name Map from the **Name Map** menu. The selections are based on the Name Maps configured in the Name Map Manager. See "Name Mapping" on page 117 for details on Name Maps. If you have

no name map selected, select the **Name Map Mgr** button to open the Name Map Manager. If you do have a Name Map selected, the button says **Open** to open that Name Map.



Missing Values

The user specifies how missing values are represented in the database. The current choices are

- **NaN.** RiverWare default, missing values are NaN in the database
- **Unchanged.** The DMI will not change missing values on import, uses NaNs on export
- **Replace With.** Provides a user input value that is substituted on import and export.

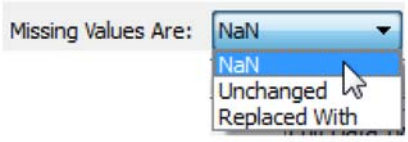


Table Slot Data

The user specifies how paired data should be imported to table slots in terms of overwriting existing data or leaving extra data unchanged.

This option is only enabled for Input DMIs. The options are:

- **Import and Resize -** a DSS DMI will import paired data by resizing the table to match the imported table. This is the default option. For example if the table slot currently has 20 rows, but the DSS data only contains 14 rows, then after import, the table slot will now have 14 rows. The 6 rows at the bottom of the table slot will be removed.
- **Import Available** - a DSS DMI will expand a table to accommodate the imported table but it won't contract a table. This leaves any extra data in the table without change. For example if the table slot currently has 20 rows, but the DSS data only contains 14 rows, then 6 rows at the bottom of the table slot will remain unchanged. Only the first 14 rows will have the newly imported data.



Units

The units that are used in the DSS database are specified in the Dataset dialog. A pull-down menu allows the user to specify that the Database DMI should **Use Database Units**, **Prefer Database Units**, or **Use Dataset Units**. Figure 4.18 shows an example.

- **Use Database Units**. The database contains all units (This option is not applicable to DSS because units are not required to be defined in DSS files, so can't be fully depended on)
- **Prefer Database Units**. Use the units in the database, if missing, use the specified Units
- **Use Dataset Units**. No units are specified in the database, use the specified units.

If you chooses either of the second two options, you must specify the units in the window below the pull-down menu. Select the ✚ icon to add a new unit type:

This adds an entry to the list. Double-click the word **NONE** under the **Type** to change the type. Figure 4.18 illustrates. Change the **Scale** and **Units** in a similar manner. These units are used to specify the units that will be written to the database if an Output DMI is used and the units that the database contains if an Input DMI is used.

Figure 4.18

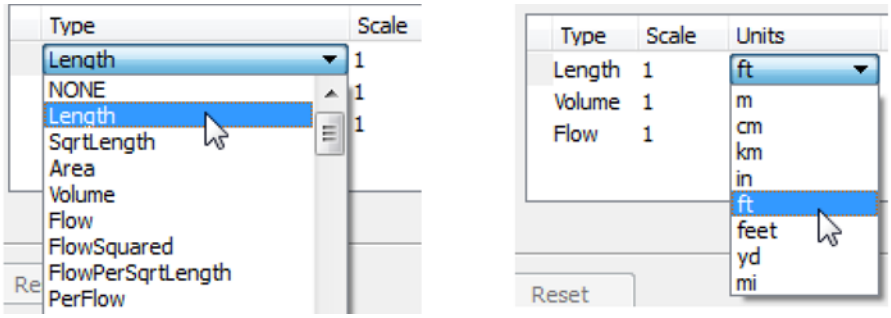
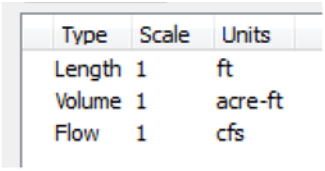


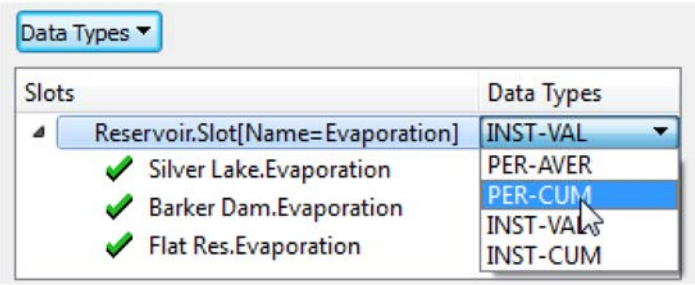
Figure 4.19 shows an example of specified units.

Figure 4.19



Data Types

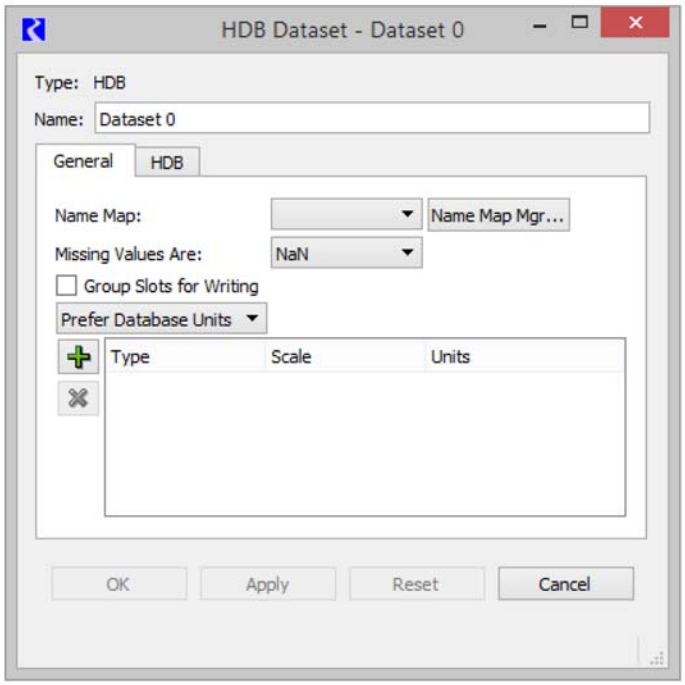
Select the **Edit Data Types** button to edit the data types. In the resulting dialog, you can specify that a selection of slots should have a certain data type. The DSS data types are: PER-VAL, PER-AVER, INST-VAL, INST-CUM. The selection is similar to other slot selections as it is dynamic and can make full use of wildcarding.



HDB Datasets

Double-clicking an HDB dataset in the Dataset Manager opens the HDB Dataset editor. In this dialog, the dataset can be renamed and configured. It has two tabs, one for general configuration and one specific to the HDB implementation.

General Tab



Name Map

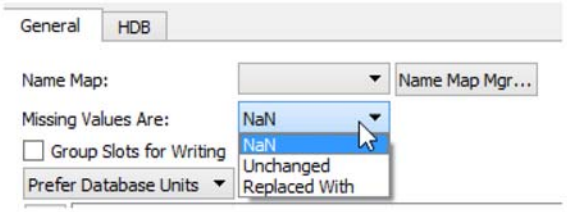
Mappings of RiverWare names to HDB IDs are contained in database tables, so Name Maps are not needed for configuring a dataset for HDB. Specifying which database mapping to use is described later in the HDB tab of the configuration dialog.

Missing Values

You specify how missing values are handled when exchanging data with HDB. In general, a NaN in RiverWare is not represented as a data record in HDB. The current choices are as follows:

- **NaN.** On export, no data record is created in HDB for NaNs in RiverWare, and any corresponding existing data record in the real tables is deleted (there would be no existing data record in the model tables). On import, missing values in HDB are made NaNs in RiverWare.
- **Unchanged**. On export, any existing data in the HDB real tables is left unchanged for a corresponding NaN value (there would be no existing data record in the model tables). On import, RiverWare data is left unchanged for a missing value in HDB.

- **Replaced With**. User provides an input value that is substituted for NaNs on export or missing values on import.



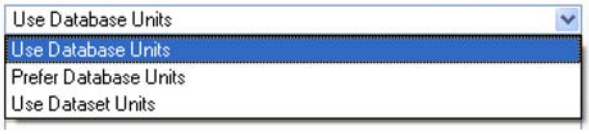
Group Slots for Writing

The **Group Slots for Writing** check box configures the DMI to group slots together when writing to the database. This option is particularly useful when you have many slots but few values in those slots. Grouping the slots minimizes the network traffic. This can significantly improve performance over a network.

Units

A menu in the Dataset dialog allows you to specify how units should be handled for a dataset. The choices are as follows:

- **Use Database Units**
- **Prefer Database Units**
- **Use Dataset Units**

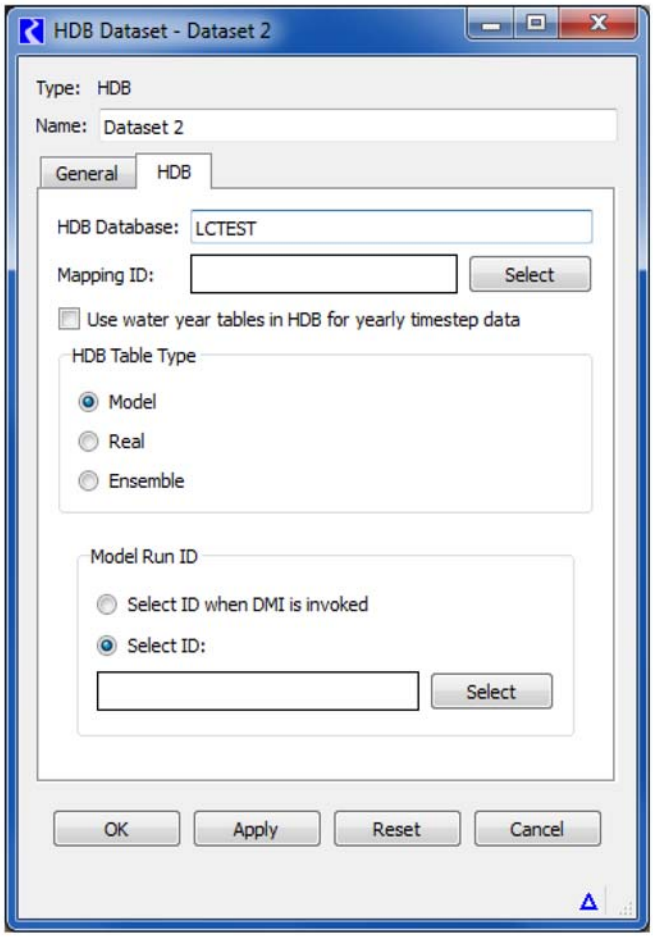


Units are defined in HDB for each piece of data, so **Use Database Units** is always applicable. If the unit type of the data in HDB does not match the unit type of the slot in RiverWare and the units cannot be converted, the slot is skipped and a warning message is given. The exception is if the unit type of the slot in RiverWare is NOUNITS. Here the unit type matching between HDB and RiverWare is skipped and the data is allowed to be exchanged with no conversions being applied.

In addition, to improve performance you can configure slot units in the Dataset and Use Dataset Units. This prevents additional calls across the network to get the units from the database.

Configuration specific to the HDB database is specified in the **HDB** tab of the HDB Dataset editor.

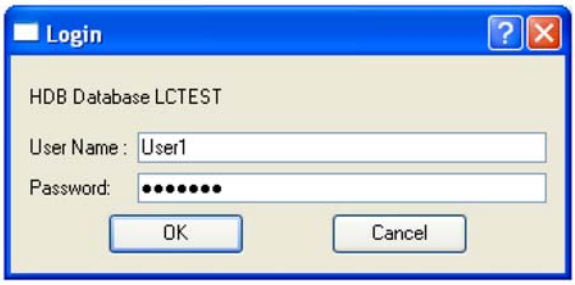
HDB Tab



HDB Database

The **HDB Database** field specifies which database the dataset will interact with. The entry in this field should be a TNS name or connect string that will allow connection through Oracle Networking Services (SQL*Net) to the desired database. You need to have Oracle networking available on your computer through Oracle Client or a full Oracle database installation.

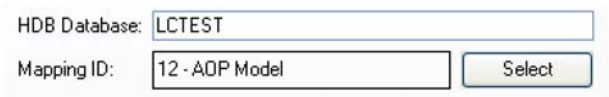
The first time in a RiverWare session that a specified database is connected to, you are prompted to provide a user name and password to log in. For a user to retrieve metadata or read and write data to HDB, they must be granted the roles model_priv_role and app_role in the database to have the necessary permissions.



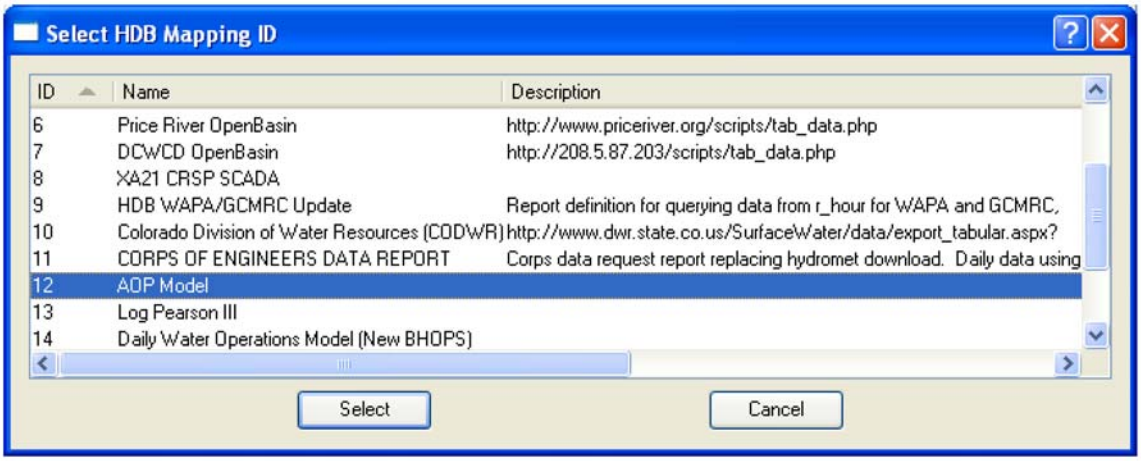
To run in batch mode, the login information must be provided in a file. The code looks for a file named.dblogins in your home directory, which is specified by the `HOME` environment variable on Solaris, and the combination of the automatically defined `HOMEDRIVE` and `HOMEPATH` environment variables on Windows. The format of the file should be a line for every database, with that line containing the database name, login, and password separated by whitespace (blanks or tabs). Blank lines and comment lines starting with # are permitted in the file. If you want to set up a different file, the `DBLOGINS` environment variable can be defined as a full path (directory and file) where the code will look for the login information.

Mapping ID

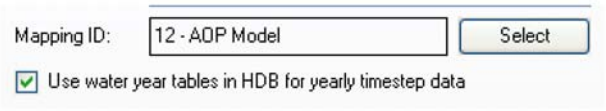
The **Mapping ID** field specifies what mapping will be used in the database to map RiverWare objects and slots to HDB sites and data types. You must select the adjacent **Select** button to pick a mapping from the database.



The **Select** button brings up the Select HDB Mapping ID dialog populated with all the available mapping IDs, their names, and descriptions from the HDB_EXT_DATA_SOURCE table in the database. You can then select a mapping, which populates the **Mapping ID** field on the HDB tab of the dataset dialog with the ID and the name of the mapping.



HDB maintains different tables for data of differing intervals (timesteps). Slots of varying timesteps can be freely intermixed in a list that is associated with a dataset, and the correct tables for each timestep will be automatically accessed in HDB. One case that does need specification is for the water year tables in HDB. There is no corresponding water year timestep in RiverWare, although some users have used slots with a 1 Year timestep to hold water year data in some of their RiverWare models. To handle this case, an option box on the dataset dialog to **Use water year tables in HDB for 1 Year timestep data** has been provided. If a user checks this box, all 1 Year timestep data read and written with the dataset will be moved to and from the water year tables in HDB instead of the year tables.

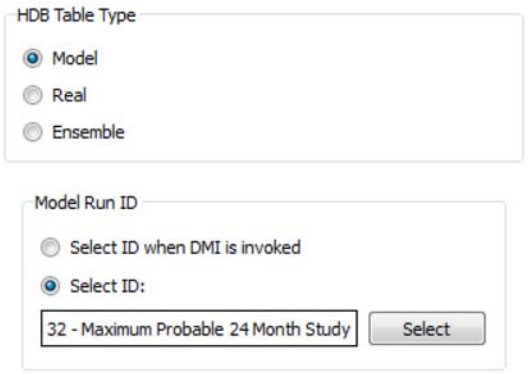


HDB Table Type

The HDB Table Type frame on the dataset dialog specifies which of the tables in HDB the dataset will interact with. The selection of **Model**, **Real**, or **Ensemble** determines the configuration options that become available in the rest of the dialog.

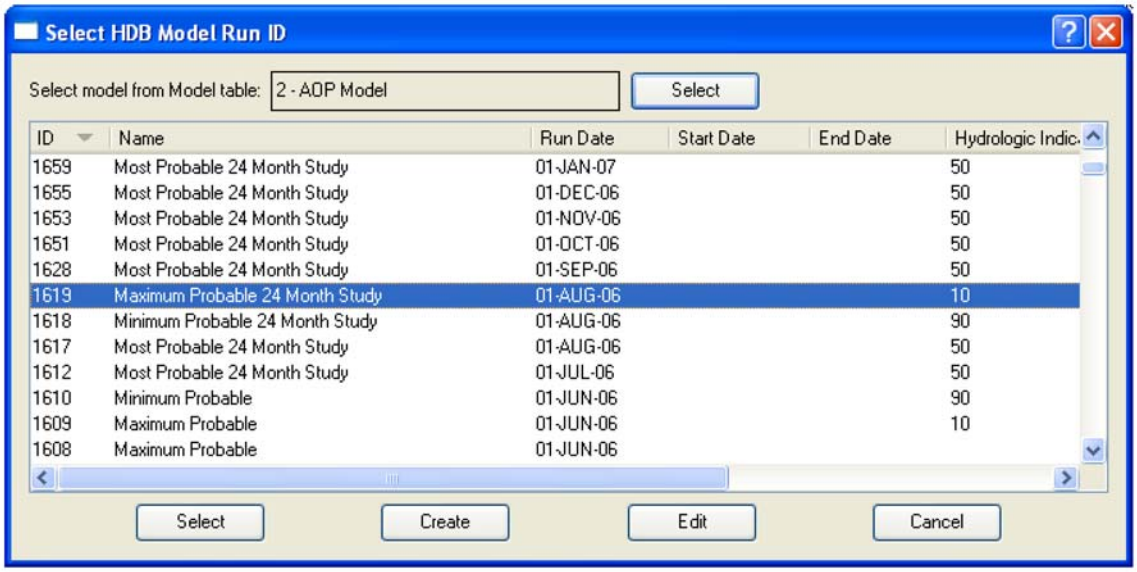
HDB Table Type—Model

A selection of **Model** in the HDB Table Type frame of the dataset dialog means that any data moved using the dataset will be read from or written to the model (m_ xxxx) set of tables in HDB. Model data is typically forecasted data, although there are no restrictions on its dates being future or historic. There can be multiple forecasts made for a piece of data at the same date and time, so data in the model tables must be associated with a model run ID that indicates from what run of a model the data originated. If data is written to a model run ID with a DMI, any previous data that was associated with that ID will be deleted from the model tables. Therefore, only the data from one particular run of a model is represented in a single model run ID. Reading or writing of model data requires metadata selection in the Model Run ID frame that appears with the selection of **Model** in the dataset dialog.

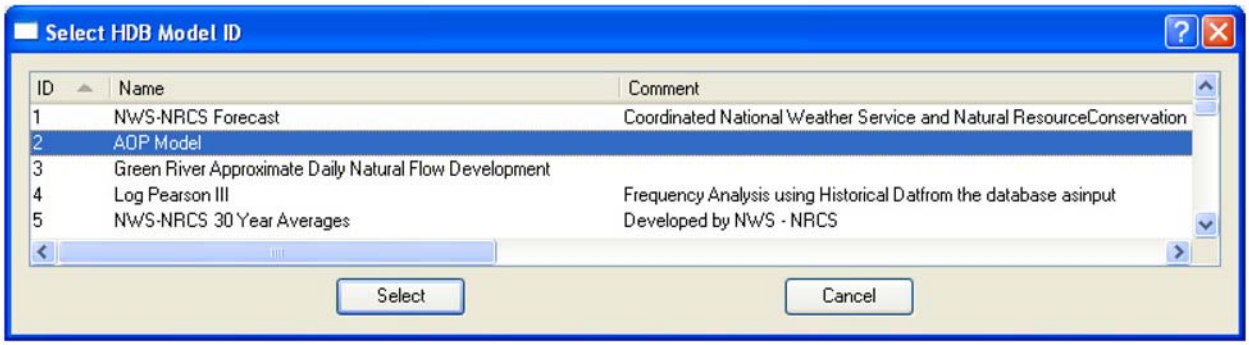


The Model Run ID frame allows you to select either **Select ID** or **Select ID when DMI is invoked**. The **Select ID** option allows you to preselect a model run ID to associate with the dataset for reading and writing data with the

model tables. To select a model_run_id, you must select the **Select** button adjacent to the model run ID field. This brings up the Select HDB Model Run ID dialog.



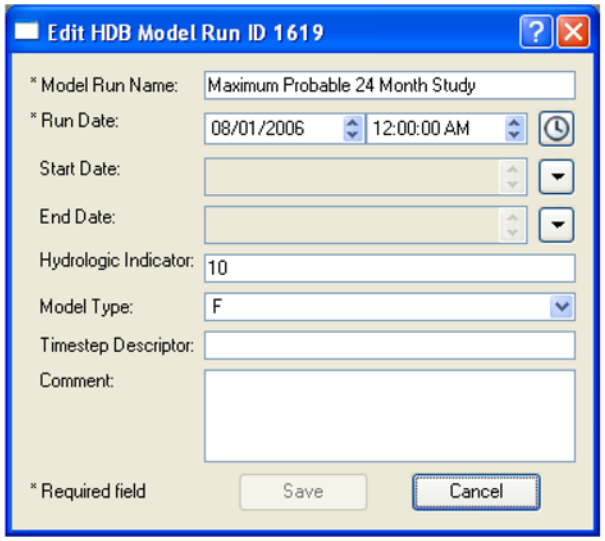
All model run IDs are associated with a particular model as defined in the database. The model run ID dialog is designed to display all the IDs in the database for a specified model. The model is selected via the **Select model from MODEL table** field at the top of the Model Run ID dialog. Selecting the **Select** button adjacent to the field brings up the Select HDB Model ID dialog populated with all the models defined in the HDB_MODEL table in the database.



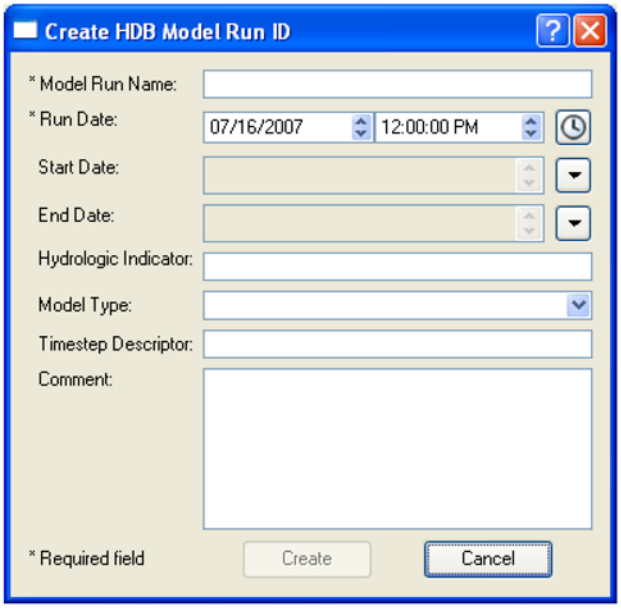
You can then select a model, which populates the **Select model from MODEL table** field on the model run ID dialog with the model ID and name, as well as populating the model_run_id list in the dialog with information for all the model_run_ids defined for the model. You can then select an existing model run ID from the list, which closes the dialog and populates the model run ID field in the Model Run ID frame of the dataset dialog with the model run ID and name.

There are two other options for Model Run IDs on the Select HDB Model Run ID dialog. Instead of just selecting an existing ID, you can also edit an existing ID or create a new one. To edit or create a model run ID you still must have selected a model at the top of the dialog. An existing ID that is displayed can then be highlighted and edited by selecting the **Edit** button at the bottom. This brings up the Edit HDB Model Run ID dialog, which displays all

the information fields for the model_run_id and allows you to edit these fields. Selecting the **Save** button will then save the edited information for this model run ID to the database.

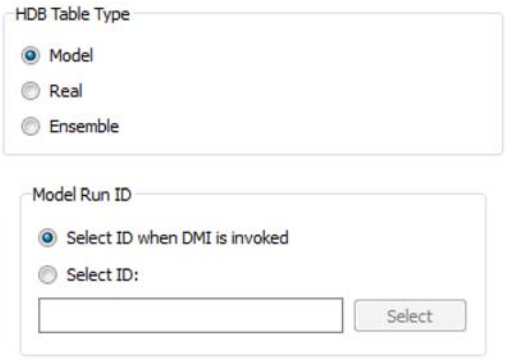


The **Create** button at the bottom of the Select HDB Model Run ID dialog allows you to create a model run ID that is associated with the model shown in the **Select model from MODEL table** field at the top of the dialog. Selecting **Create** brings up the Create HDB Model Run ID dialog, which contains fields where information can be entered for the new model run ID. The **Run Date** field is populated with the start time of the run, but can be changed by selecting any portion of the date and typing or using the spinners. At least the required fields marked with an asterisk must be filled out. Selecting the **Create** button will then save entered data to the database as a new model run ID where it will be assigned the next available model run ID number. The new ID will then be added to the list and highlighted back on the Select HDB Model Run ID dialog.



The Model Run ID frame of the dataset dialog has a second option for specifying model run IDs by selecting **Select ID when DMI is invoked**. When this option is selected, you do not preselect a model run ID that is always

used with the dataset, but rather you are prompted for one when a DMI containing the dataset is executed. This option could be useful in a case where the DMI is run to record forecast data on a regular basis with the data always being preserved in separate model run IDs in the model tables. To limit the potential for you to accidentally overwrite the previously saved data by forgetting to change the ID; you must select the ID each time the DMI is run.



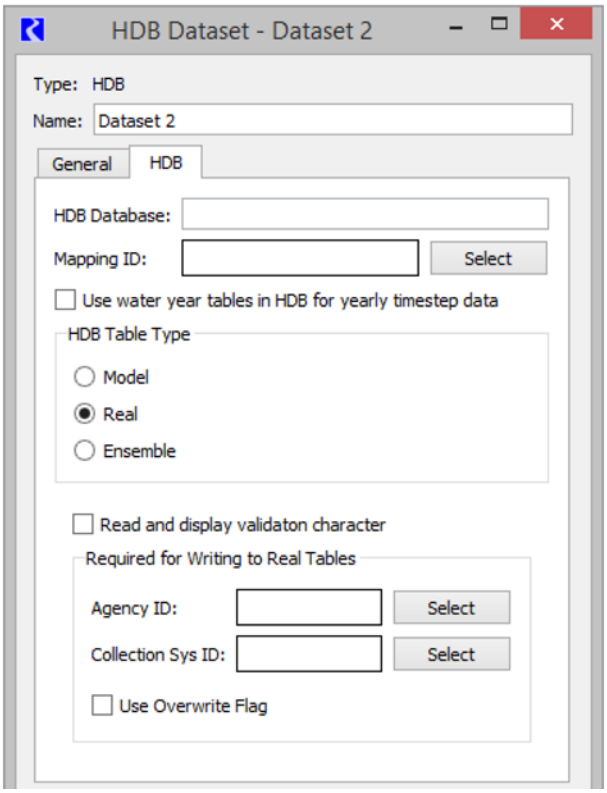
Selecting an ID under the **Select ID when DMI is invoked** option utilizes the same dialogs as does the preselect ID option. The Select HDB Model Run ID dialog appears where you can select a model and see its existing IDs. An existing ID can then be selected, an existing ID can be edited and then selected, or a new model run ID for the model can be created then selected.

> **Note:** If the same dataset is used multiple times in a DMI, you are only prompted once to associate a model run ID with the dataset; the same dataset cannot be used with multiple model run IDs in a single DMI or DMI group execution.

HDB Table Type—Real

A selection of **Real** in the Table Type frame means that any data moved using the dataset will be read from or written to the real (r_ xxxx) set of tables in HDB. Real data are typically current or historical observed values;

future data cannot be loaded into the real tables. A write of real data to HDB requires that some other metadata be selected by you in the Real frame that appears with the selection of a **Real** table type in the Dataset dialog.



Validation Characters

HDB has a per-value Validation field in the R_DAY, R_HOUR and R_INSTANT tables which typically indicates whether the data is provisional or approved. The field is a 1 byte character which is usually a “P” for provisional and an “A” for approved.

The **Read and display validation character** option allows you to import validation characters from HDB into Notes on Series Slot, which are then shown on slot and SCT dialogs. The Notes Manager dialog enables you to easily see all slots and timesteps with a particular note, for example to easily see all provisional slots and timesteps. See “Notes on Series Slots” in *User Interface* for details on Notes.

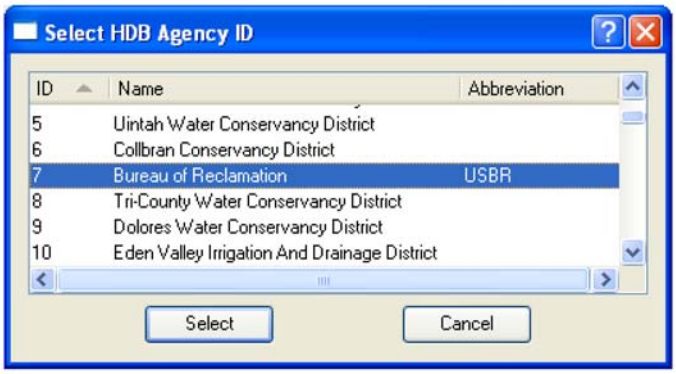
When the **Read and display validation character** is checked, the Validation field values are also imported through the DMI and set on a Series Slot Note, associated with the particular slot and timestep. The Series Slot Note Group which holds the validation characters is named “HDB Validation Group”. The group's name is fixed; you can't edit it. In the Edit Note Group, the label above the name is Fixed Note Group Name and the name is read-only. You can edit the group's color.

Required For Writing to Real Tables

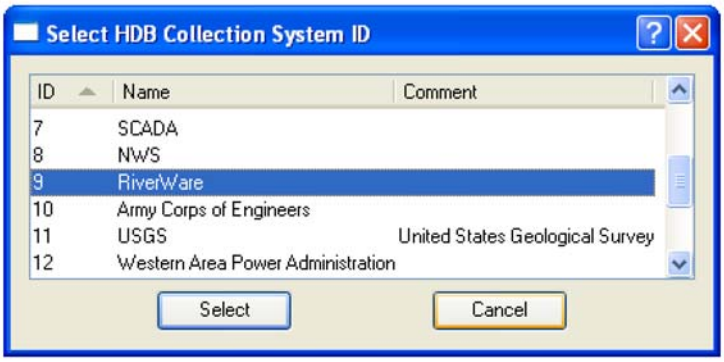
The remainder of the controls for the Real table type are described in this topic.



The **Agency ID** field specifies what agency ID will be associated with the data when it is written to the real tables. You must select an agency ID by selecting the **Select** button adjacent to the field. This will bring up the Select HDB Agency ID dialog populated with all the available agency IDs, their names, and abbreviations from the HDB_AGEN table in the database. You can then select an agency, which populates the **Agency ID** field on the dataset dialog with the ID and name of the agency.



The **Collection Sys ID** field specifies what collection system ID will be associated with the data when it is written to the real tables. You must select a collection system ID by selecting the **Select** button adjacent to the field. This will bring up the Select HDB Collection System ID dialog populated with all the available collection system IDs, their names, and comments from the HDB_COLLECTION _SYSTEM table in HDB. You can then select a collection system, which populates the **Collection Sys ID** field on the dataset dialog with the ID and name of the collection system.

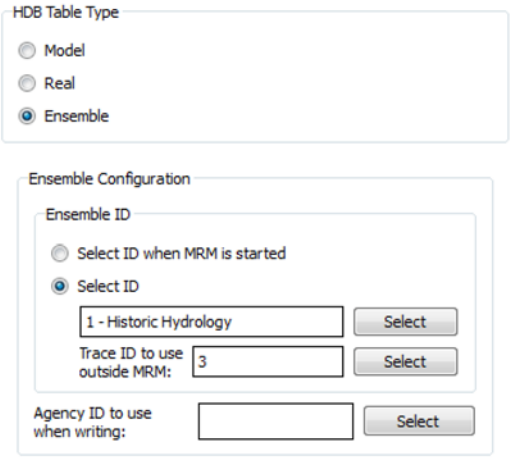


The **Use Overwrite Flag** check box specifies whether or not data is written to the real tables with an overwrite flag. This flag has some special meaning for how the data is handled in the database, namely that when the value is

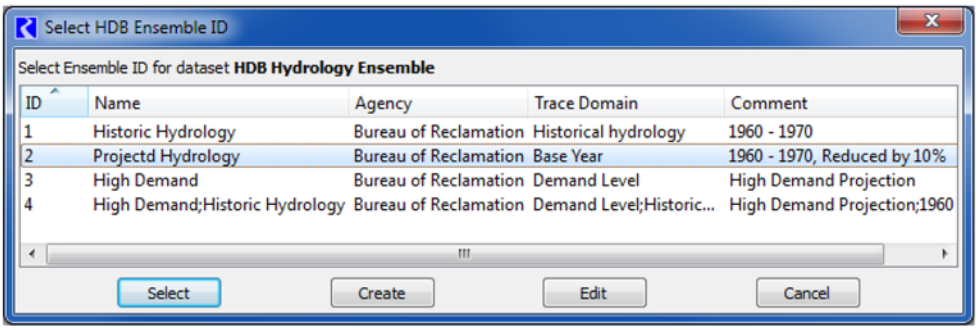
written to its real table, it cannot be overwritten by aggregations of data from a shorter interval. Normally this box would not be checked.

HDB Table Type—Ensembles

A selection of **Ensemble** in the Table Type frame means that any data moved using the dataset will be determined through the REF_ENSEMBLE table in the database. An ensemble has a number of traces associated with it, and each trace has an associated model run ID. An ensemble would be used with MRM so that each run of the multiple run would correspond to a trace in the ensemble, and the data for that trace would reside under the trace's model run ID in the model (m_ xxxx) set of tables in HDB. See "Ensemble Configuration" in *Solution Approaches* for details.

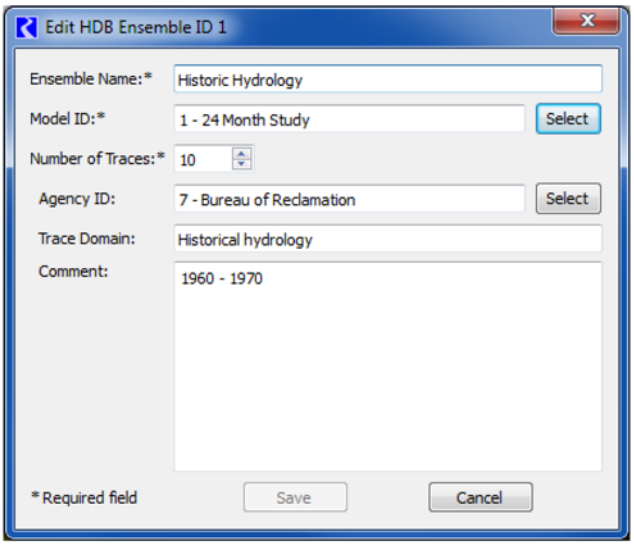


The Ensemble ID frame has options for you to **Select ID** or **Select ID when MRM is started**. The **Select ID** option allows you to preselect an ensemble ID to associate with the dataset for reading or writing data with the model tables. You must select an ensemble ID by selecting the **Select** button adjacent to the ensemble field. This will bring up the Select HDB Ensemble ID dialog populated with all the available ensemble IDs, their names, the agency, the domain for the traces, and comments from the REF_ENSEMBLE table in the database. You can then select an ensemble, which populates the **Ensemble ID** field on the dataset dialog with the ID and name of the ensemble.

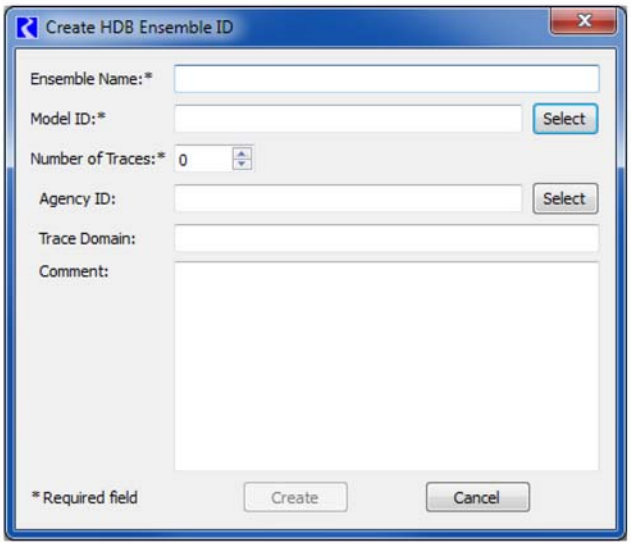


There are two other options for ensemble IDs on the Select HDB Ensemble ID dialog. Instead of just selecting an existing ID, you can also edit an existing ID or create a new one. An existing ID that is displayed can be highlighted and edited by selecting the **Edit** button at the bottom. This brings up the Edit HDB Ensemble ID dialog,

which displays all the information fields for the ensemble id and allows you to edit these fields. Selecting the **Save** button will then save the edited information for this ensemble ID to the database.

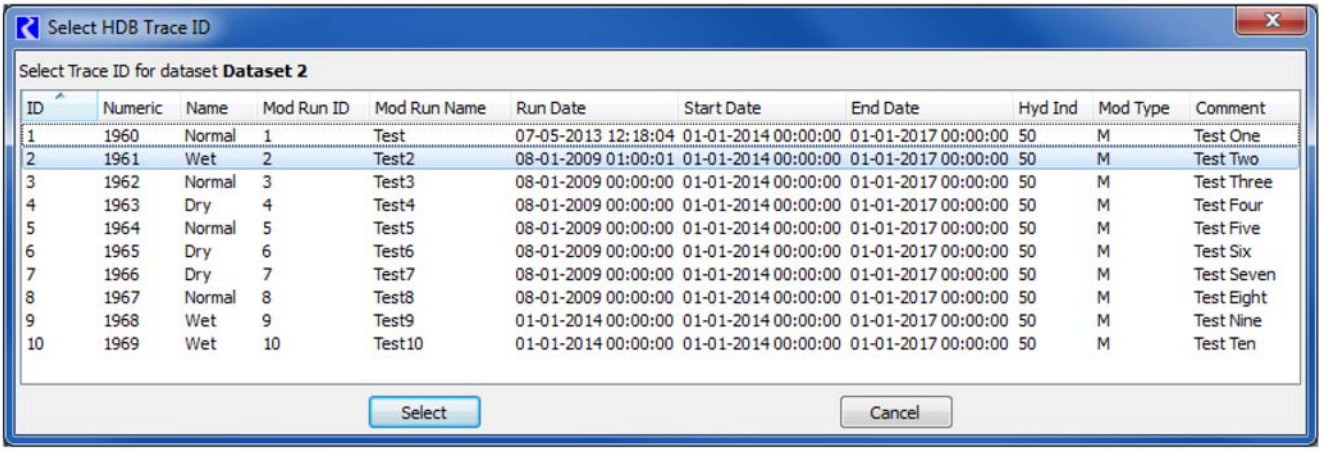


The **Create** button at the bottom of the Select HDB Ensemble ID dialog allows you to create an ensemble ID. Selecting this brings up the Create HDB Ensemble ID dialog, which contains fields where information can be entered for the new ensemble ID. At least the required fields marked with an asterisk must be filled out. Selecting the **Create** button at the bottom will then save entered data to the database as a new ensemble ID. The new ID will then be added to the list and highlighted back on the Select HDB Ensemble ID dialog.



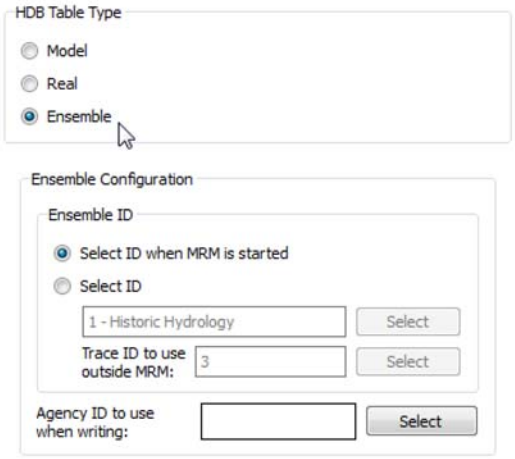
It may be convenient to use an ensemble dataset outside of a multiple run to pull a particular trace from the ensemble for use in a single run or to manually test a DMI. Selecting the **Select** button adjacent to the **Trace ID to Use Outside MRM** field will open the Select HDB Trace ID dialog that is populated with all the traces for your selected ensemble ID specified for the dataset. The dialog shows descriptive information for each trace from the

REF_ENSEMBLE_TRACE table and also information for the trace's model run ID from the REF_MODEL_RUN table. You can then select a trace, which populates the trace ID into the dataset dialog.



ID	Numeric	Name	Mod Run ID	Mod Run Name	Run Date	Start Date	End Date	Hyd Ind	Mod Type	Comment
1	1960	Normal	1	Test	07-05-2013 12:18:04	01-01-2014 00:00:00	01-01-2017 00:00:00	50	M	Test One
2	1961	Wet	2	Test2	08-01-2009 01:00:01	01-01-2014 00:00:00	01-01-2017 00:00:00	50	M	Test Two
3	1962	Normal	3	Test3	08-01-2009 00:00:00	01-01-2014 00:00:00	01-01-2017 00:00:00	50	M	Test Three
4	1963	Dry	4	Test4	08-01-2009 00:00:00	01-01-2014 00:00:00	01-01-2017 00:00:00	50	M	Test Four
5	1964	Normal	5	Test5	08-01-2009 00:00:00	01-01-2014 00:00:00	01-01-2017 00:00:00	50	M	Test Five
6	1965	Dry	6	Test6	08-01-2009 00:00:00	01-01-2014 00:00:00	01-01-2017 00:00:00	50	M	Test Six
7	1966	Dry	7	Test7	08-01-2009 00:00:00	01-01-2014 00:00:00	01-01-2017 00:00:00	50	M	Test Seven
8	1967	Normal	8	Test8	08-01-2009 00:00:00	01-01-2014 00:00:00	01-01-2017 00:00:00	50	M	Test Eight
9	1968	Wet	9	Test9	01-01-2014 00:00:00	01-01-2014 00:00:00	01-01-2017 00:00:00	50	M	Test Nine
10	1969	Wet	10	Test10	01-01-2014 00:00:00	01-01-2014 00:00:00	01-01-2017 00:00:00	50	M	Test Ten

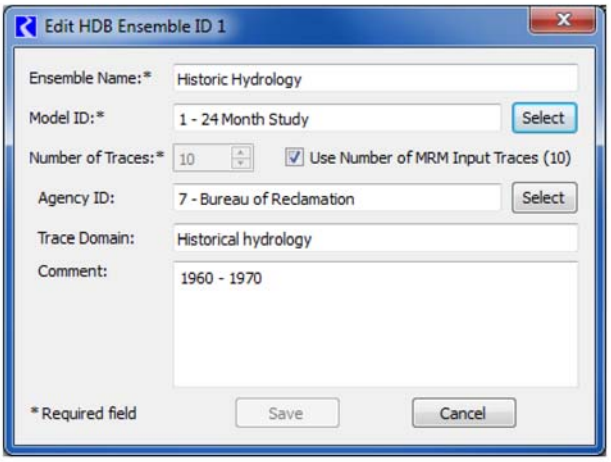
The Ensemble ID frame of the dataset dialog has a second option for specifying ensemble IDs by selecting **Select ID when MRM is started**. When this option is selected, you do not preselect an ensemble ID that is always used with the dataset, but rather you are prompted for one when an MRM run containing this dataset in an input or output ensemble is started. This option could be useful, for example, where you want to select a different output ensemble for each execution of the multiple run so that data from earlier runs is not overwritten.



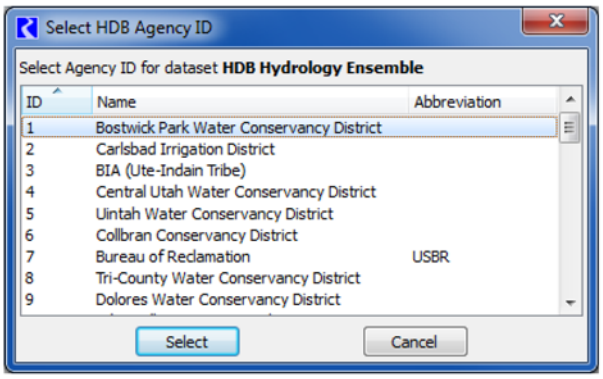
Selecting an ID under the **Select ID when MRM is started** option utilizes the same dialogs as does the preselect ID option. The Select HDB Ensemble dialog appears when you start the multiple run where you can see existing ensemble IDs. An existing ID can then be selected, an existing ID can be edited and then selected, or a new ensemble ID for can be created then selected.

> **Note:** If an ensemble id is being chosen for an output ensemble, the edit and create ensemble dialogs in this case have an additional trace control to **Use Number of MRM Input Traces (xx)**, where xx indicates that number of traces. This provides an easy way for the edited or newly created output

ensemble to contain the correct number of traces to hold the run data as defined by the input ensembles.



In addition to the ensemble id options discussed above, there is also an option under the Ensemble Configuration frame to select an **Agency ID to use when writing**. Selecting **Select** will open the Select HDB Agency ID dialog where one of the agency ids defined in HDB can be chosen. If chosen, that agency id will be written to the dataset's ensemble when that dataset is executed as part of an output ensemble. If one is not chosen, the behavior will be to preserve any agency id that is already defined in the database for the output ensemble.



An HDB ensemble can have metadata that describe the ensemble as well as metadata that describe each trace. Metadata is a keyword associated with a value, such as "comment"/"Historic Hydrology". In the HDB database, values for the "domain", and "comment" keywords are held in the trace_domain, and cmmt fields of the REF_ENSEMBLE table. Other keyword/value pairs to describe ensembles are held in the REF_ENSEMBLE_KEYVAL table.

For trace metadata, the values for the "name" and "numeric" keywords are held in the trace_name and trace_numeric fields of the REF_ENSEMBLE_TRACE table. Other keywords to describe traces are held in the REF_MODEL_RUN_KEYVAL table.

See "Ensemble Configuration" in *Solution Approaches* for details on using Ensembles in MRM.

Excel Datasets

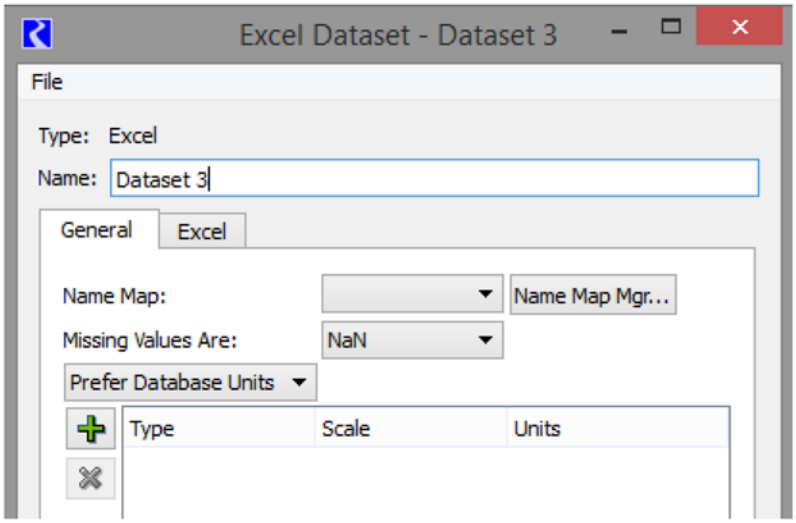
Excel datasets provide user configuration options to specify how you wish the Database DMI to work with Microsoft Excel. Excel datasets can only be used on the Windows platform and Excel must be available on the system. RiverWare will start a non-visible copy of Excel in the background and access the specified workbook during DMI execution. As a result, there are two important things to remember:

> **Note:** You should not have the specified workbook open in a different copy of Excel as this may lock out RiverWare from accessing the workbook.
>
> **Note:** RiverWare is operating on the Excel file, so you will not see any unsaved changes to the file that may be made in a different Excel session.

See "Example 2: Excel DMI to Export Data Using Headers" on page 74 for a use case of creating a DMI to Excel.

General Tab

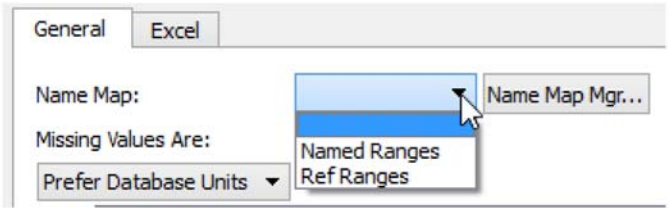
Double-clicking an Excel dataset in the Dataset Manager opens the Excel Dataset editor. In this dialog, the dataset can be renamed and configured. It has two tabs, one for general configuration and one specific to the Excel implementation. Following is a description of the configuration options.



Name Map

Select the desired Name Map from the **Name Map** menu. The selections are populated based on the Name Maps configured in the Name Map Manager. See "Name Mapping" on page 117 for details on creating Name Maps.

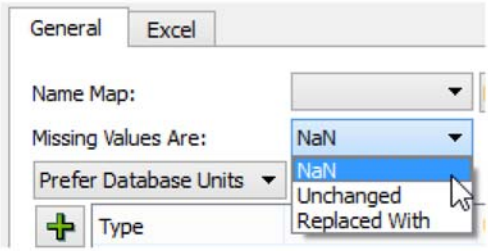
Name maps have special relevance for moving data using range specifications and header specification. See "Slot Mapping—Spreadsheet Ranges Approach" on page 105 and "Using Name Maps" on page 109 for details.



Missing Values

The user specifies how missing values are handled with Excel. The current choices are

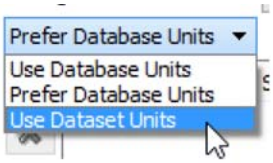
- **NaN.** NaN value in RiverWare is written as NaN to Excel, NaN or blank value in Excel is read as NaN into RiverWare
- **Unchanged**. The DMI will not change missing values on import, uses NaNs on export
- **Replaced With**. Provides a user input value that is substituted on import and export



Units

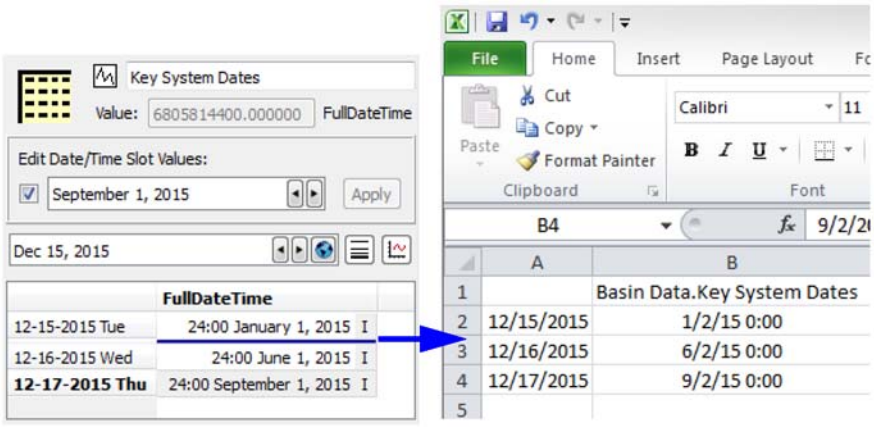
The units that are used in interacting with Excel are specified in the Dataset dialog. A pull-down menu allows the user to specify that the Database DMI should **Use Database Units**, **Prefer Database Units**, or **Use Dataset Units**.

- **Use Database Units**. Since Excel contains no units associated with a value in a cell, the slots' units (i.e. from the Unit Scheme) as displayed in RiverWare are substituted for Database Units and will be used for import and export under this option
- **Prefer Database Units**. For Excel, this option uses the slots' units (i.e. from the Unit Scheme) in RiverWare unless a unit is specified in the Dataset, then the specified unit is used
- **Use Dataset Units**: Uses the units specified in the dataset only



See "Units" on page 84 for instructions on specifying the units to use.

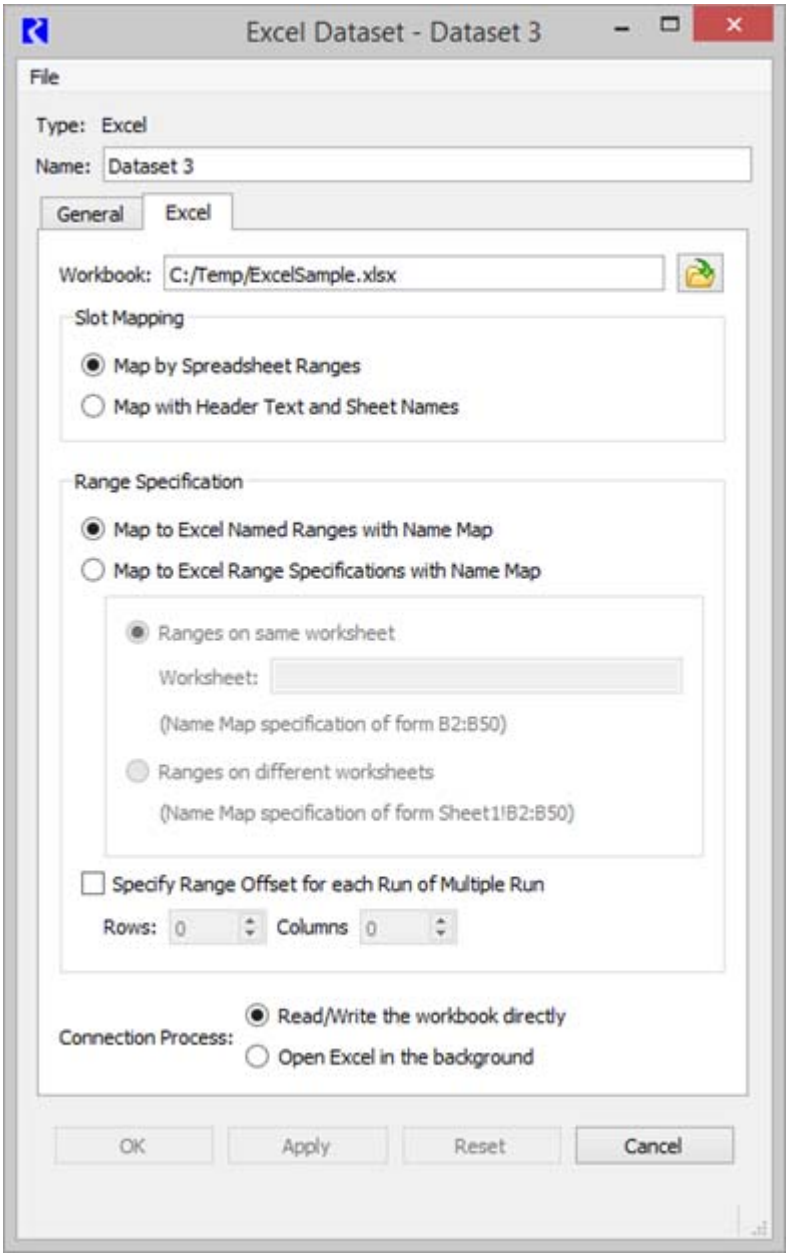
Note: For import and export of series slots that have the DateTime unit type, the DMIs translate to and from the Excel date representation. This allows dates to display correctly in both RiverWare and in Excel. But, Days/months/years in RiverWare are at the end of the timestep, which would be represented in Excel at the beginning of the next timestep. For example 24:00 June 1, 2015 in RiverWare would translate to 00:00 June 2, 2015 in Excel. Also, partial datetimes and dates before 1900 are negative values in Excel, so are just displayed with # signs as out of range. However the representation is still consistent and these date values will round trip correctly when exported to Excel and imported back into RiverWare.



Note: Text Series Slot values can be imported or exported using Excel Datasets and Database DMIs. These behave just like series slots but import or export the text strings. Be aware that a NaN in Excel will be interpreted as a string, not as an invalid value. Use blank cells in Excel to prevent importing NaN text strings. See "Text Series Slots" in *User Interface* for details.

Excel Tab

Configuration specific to Excel is specified in the **Excel** tab of the Excel Dataset editor.



Workbook

In all cases, the Workbook that the dataset will read data from or write data to must be specified. The path and workbook name can be typed into the **Workbook** text box, or you can select the adjacent button to select the file. When executing an input DMI, this workbook must already exist. For an output DMI, the workbook will be created if it does not already exist.

Slot Mapping

Two approaches are provided for mapping data between RiverWare and Excel.

- Ranges: Specify the Excel range for each RiverWare slot. Choose the **Map by Spreadsheet Ranges** approach,
- Headers: Use a row header, a column header, and a worksheet name to map the Excel data with a RiverWare slot. The **Map by Header Text and Sheet Name** approach.

Table 4.3 compares these approaches. Then more information is presented on the configuration options for both the Range and Header approach.

Table 4.3 Comparing Excel dataset approaches

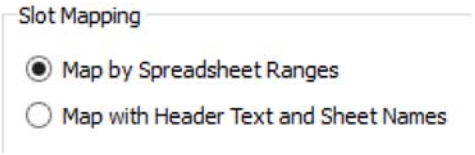
Question	Ranges Approach	Headers Approach
What is the format of the Excel data?	Your data is in any format in Excel, but you must then specify how the data maps to RiverWare slots.	Your data in Excel has row header names, column header names, and sheet names that identify the data in a way that matches one of the orientations in the dataset.
How is the data mapped?	A DMI Name Nap specifies how the data in RiverWare slots “maps” to data in an Excel sheet. Name Maps can specify an Excel absolute cell reference (A1:A23) or a named range.	The text in header cells in Excel and the name of the sheet “map” the data to RiverWare slots.
Can I have more timesteps of data in Excel than I want to import/export?	Yes, but the mapped Excel range must match the number of timesteps specified in the DMI. That is, the size of the range you specify must exactly match the number of values in the DMI.	Because this approach includes timestep information, the Excel sheet can have more data in it than you are processing via the DMI
Can I import and export table, periodic, or scalar slots?	Yes, your named ranges or cell references can map to non-series data.	Yes, a dataset using the Header approach can either deal with Series slots OR Table, Periodic or Scalar slots, but not both at the same time.
Can or do I use Name Maps?	You must use name maps to specify where the data will import from/export to in Excel.	Name Maps can optionally allow you to have different names of objects or slots in RiverWare than the header text in Excel.
How are MRM runs handled?	Data for each MRM run is offset from the previous run by a certain number of rows and columns, starting with the mapped range.	MRM runs is one of the variables you specify in the Excel orientation. MRM runs can be on sheets, columns or rows. Their names can optionally be Run0, Run1, Run2, etc. or trace number as set up in the MRM configuration (i.e. Trace5, Trace6, Trace 7, etc.)
How do I specify the Excel sheet?	Either specify a single sheet name or include the sheet name in the Name Map.	Sheet names depend on the selected orientation and can be run, slot, or timestep names.
What is the format of the timestep in Excel?	No timestep information is read or written.	You specify to either use the end of timestep (E.g. Jan, 1, 2011 23:59) or the beginning of timestep format (E.g. Jan 1, 2011 0:00).

Table 4.3 Comparing Excel dataset approaches (Continued)

Question	Ranges Approach	Headers Approach
How do I specify the units for Excel?	No unit information is read or written. Units in Excel are assumed to be the slot's units in RiverWare (i.e. from the active Unit Scheme). The General tab of the dataset can be used to specify a mapped unit by type if the units in Excel are different than the RiverWare slot units.	You can optionally write the unit name with the slot name to Excel or can read a slot name from Excel if it has a unit name with it. However the assumed units for values and the possibility for mapped units work the same as in the Range approach.
In general, why would I use this approach?	This approach is more work because a map entry must be created in a name map for every slot. Use this format if you are taking the data out of or inserting it into a spreadsheet that does not have the header information and if the amount of data is constant. Perhaps you have an existing spreadsheet that cannot be changed.	This approach is recommended for series data, particularly if you want to import or export data that will change length over time. For example, you have an Excel sheet with 100 years of daily data, but your model is for the current year and is updated every year. This approach is also very useful for Table, Periodic and Scalar slots that can be represented in the spreadsheet in the required format.
Where can I find more information	"Slot Mapping—Spreadsheet Ranges Approach" on page 105	"Slot Mapping—Header Text and Sheet Names Approach" on page 109

Slot Mapping—Spreadsheet Ranges Approach

The Map by Spreadsheet Ranges approach presents several options for specifying ranges associated with slots. In all cases, the Name Map functionality of Database DMIs is used to map individual slots to their range specifications. You can map Series, Table, Periodic, or Scalar slots using this approach.



Note: The size of the range specified must match the number of values to be read or written with the DMI. No header or other cells should be included in the range.

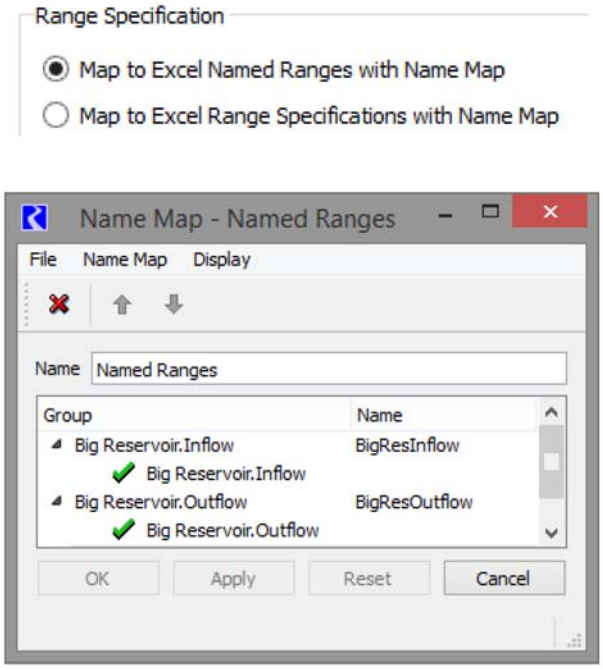
Note: Object name maps are not applicable to the range approach and are ignored. In this approach, the Excel range must be specified for each slot using a slot name map.

Discussions of the range specification options along with example Name Maps follow.

Map to Excel Named Ranges

Under the **Map to Excel Named Ranges with Name Map** option, the user would set up named ranges in their Excel workbook. Named ranges are a feature of Excel where a number of cells can be selected and assigned a name by the user to identify these cells. The Name Map created in RiverWare and selected in the **General** tab of the

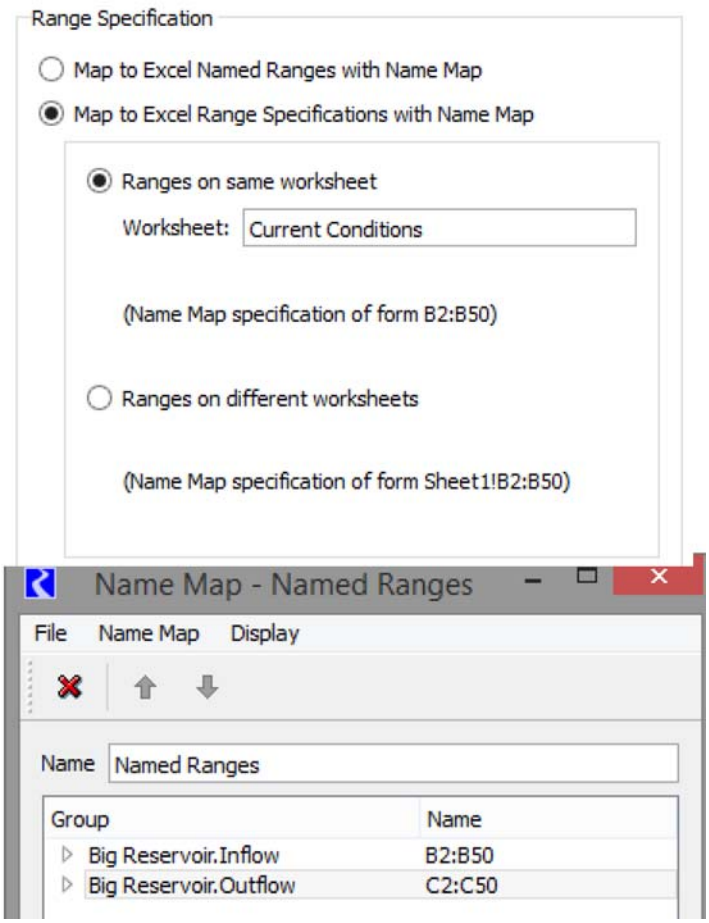
dataset dialog then will associate the RiverWare slot with the named range as seen in the following example name map. In this case, BigResInflow and BigResOutflow are named ranges in the Excel sheet.



Map to Excel Range Specifications

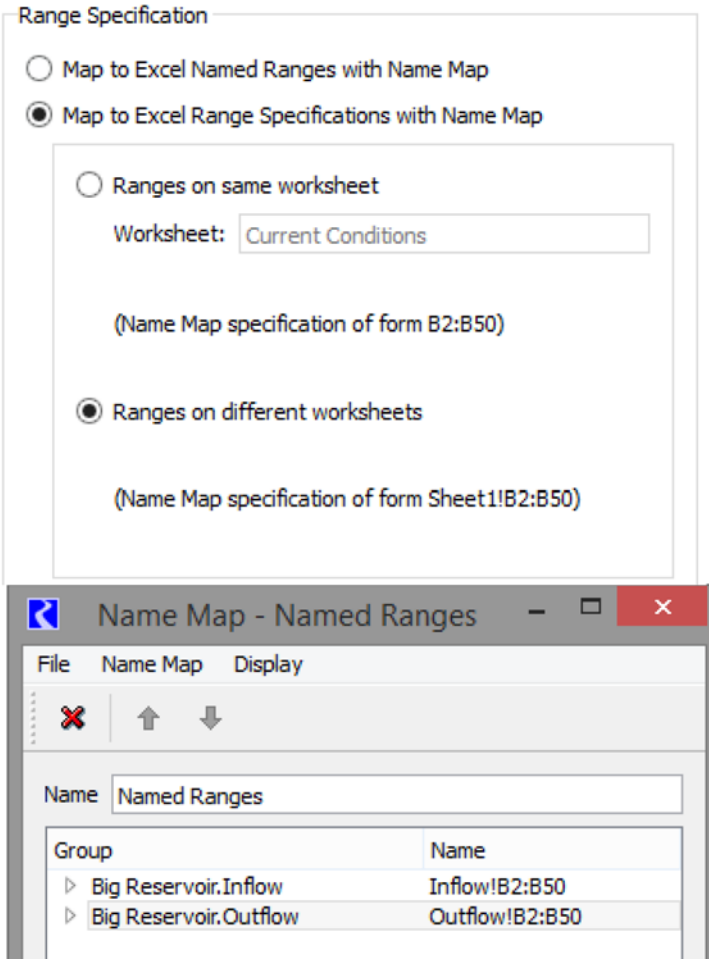
In the **Map to Excel Range Specifications with Name Map** option, there are two ways to specify ranges for slots. If all the ranges for the slots are on the same worksheet, then the **Ranges on same worksheet** option can be used. Here the worksheet name is specified in the **Worksheet** text box. Ranges specified in the name map can be of the form B2:B50, meaning the range will be from cell B2 to cell B50 of the specified worksheet. Figure 4.20 shows an example of the type of Name Map created and selected on the **General** tab of the dataset dialog for this approach.

Figure 4.20



If ranges for the slots are on different worksheets then the **Ranges on different worksheets** option must be used. Under this option, ranges specified in the name map must include the worksheet name as part of the range specification in the form Sheet1!B2:B50. In this way, ranges referring to different worksheets can be included in the same name map. Figure 4.21 shows an example of this type of Name Map.

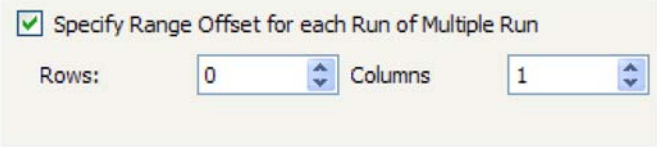
Figure 4.21



Specify Range Offset

Under all variations of the Map by Spreadsheet Ranges approach, there is an option available to Specify Range Offset for each Run of Multiple Run. In a multiple run, the user may want to read different data into a slot for each run or write out the results for the slot under each run. If the option is checked, the mapped range for the slot is used for the first run, and the specified row and column offset is applied for each run thereafter. For example, the range for a slot specified in the dataset's name map could be Sheet1!B2:B50 and the offset specified here could be 0 rows and 1 column. For an input DMI in a multiple run, the first run will read range B2:B50 on Sheet1 into the slot, the second run will offset one column and read range C2:C50 on Sheet1, the third range D2:D50, etc. For an output

DMI, the slot's results from the first run will be written to range B2:B50 on Sheet1, results from the second run will offset one column and be written to range C2:C50 on Sheet1, the third to range D2:D50, etc.



Slot Mapping—Header Text and Sheet Names Approach

The Map by Header Text and Sheet Name approach is the alternative to using ranges to move RiverWare slot data to or from Excel. The approach uses a row header, a column header, and a worksheet name to map Excel data with a RiverWare slot. Figure 4.22 shows an example of the dataset dialog with this configuration selected.

> **Tip:** When setting up an input DMI using this Excel dataset approach (especially the first time), it is often easiest to configure the DMI as an *Output* DMI first, run the DMI to create an Excel spreadsheet with the right formatting and desired structure. Edit that spreadsheet or use it as a guide for setting up your data in that format. Then, configure the DMI to be an *Input* DMI that will bring your data into RiverWare.

Using Name Maps

Name maps are not needed in the header approach. However they can be used to map to a different name in Excel. The following is performed for each slot:

- If there is no name map, the RiverWare object and slot name is used in Excel.
- If there is a slot mapping only, the specified mapped name is used in Excel.
- If there is an object mapping only, the mapped object is used for the object name followed by a period and the RiverWare slot name. This combined string is used in Excel.
- If there is a slot *and* object mapping, the mapped object name is used followed by a period and then the mapped slot name. This combined string is used in Excel.

Slot Types

An Excel dataset using the header approach can either import/export Series Slot data or Table / Periodic / Scalar Slot data, but not both. Choose the desired slots that will be accessed by this dataset.

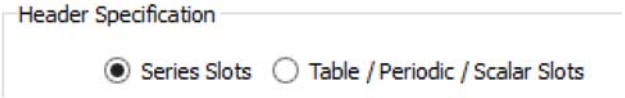
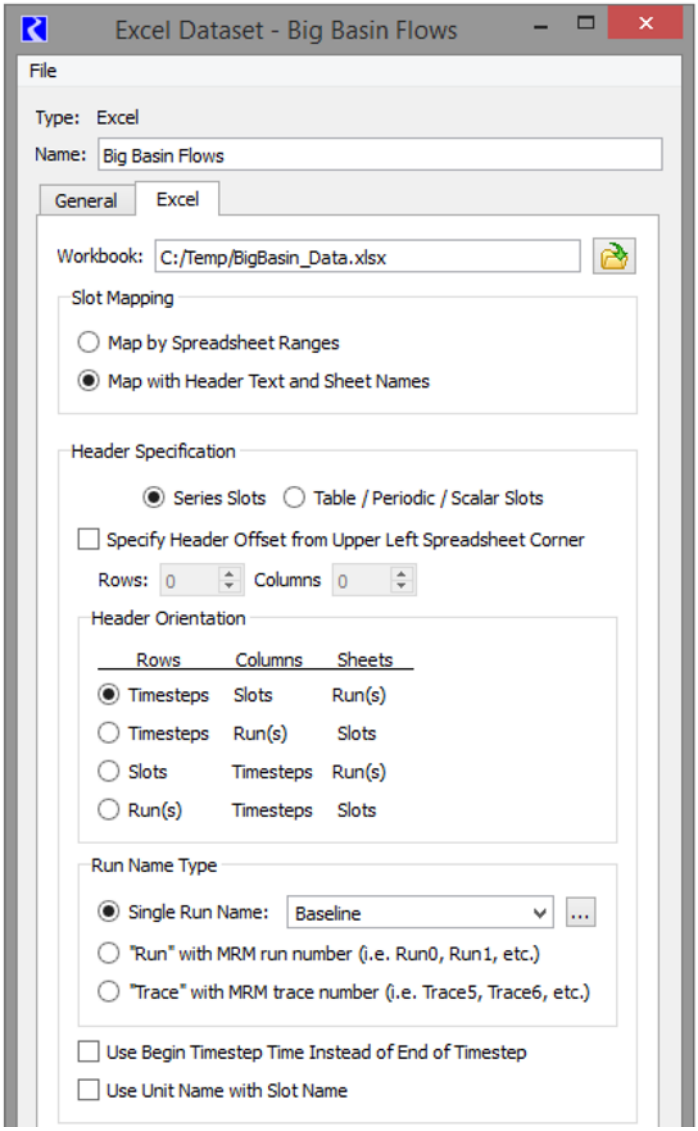


Figure 4.22



Series Slots Configuration Options

When the Series Slots option is chosen, you can specify the format of the series slots in the spreadsheet.

The Header Orientation frame provides four orientation choices for how timesteps, slots, and runs will map to rows, columns, and sheets in Excel. For example, the first orientation as selected in the adjacent screenshot indicates that rows are timesteps, columns are slots, and worksheets are runs.

Note: Orientations for putting each timestep on a separate sheet are not supported.

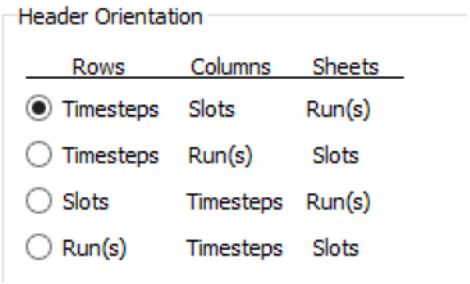


Figure 4.23 shows an Excel spreadsheet with the first orientation. In this orientation, timesteps are rows, so the timestep header is the first column with a time label for each row. Similarly, the slot header is the first row with each column labeled with the slot name. Sheets are runs, so the sheet is named with the single run name specified in the dialog.With the orientation specified, the data in Excel associated with a slot over a time range can be found for reading via an input DMI, or can be found and overwritten, if existing, or be newly created by writing via an output DMI.

Figure 4.23

	A	B	C
1		Storage Reservoir.Inflow	Storage Reservoir.Outflow
2	1/2009	1000	800
3	2/2009	1100	800
4	3/2009	1200	800
5	4/2009	1300	1000
6	5/2009	1400	1300
7	6/2009	1500	1500
8	7/2009	1450	1500
9	8/2009	1300	1500
10	9/2009	1200	1400
11	10/2009	1000	1200
12	11/2009	900	1200
13	12/2009	800	1000
14	1/2010	600	1000
15	2/2010	500	800
16	3/2010	500	700
17	4/2010	500	600

Baseline Scenario | Sheet2 | Shee

Note: On subsequent executions of an output DMI using the header approach, if there is already a matching header, data will be overwritten. If there is not already a matching header, the new header and data will be appended to any existing data. So in the example spreadsheet above, if you change the output slot from StorageResevoir.Outflow to Storage Reservoir.Storage and rerun the DMI (without deleting the.xlsx file), it will replace the inflow column with any new data and will append a StorageReservoir.Storage column after the Outflow column (i.e. it won't replace the outflow column with storage, but will instead append it).

The headers may not always be the first row and the first column. To accommodate these cases, an option is provided to specify how the headers are offset from the corner of the spreadsheet. By indicating how many rows and how many columns the header is offset from the upper left corner of the spreadsheet, the user can match how data resides in an existing spreadsheet or indicate where the headers and data should be written to a new sheet.

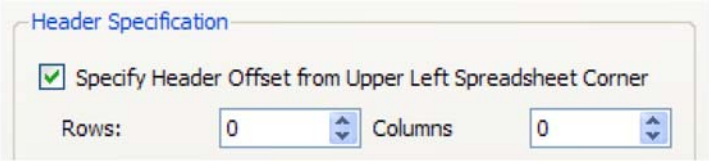
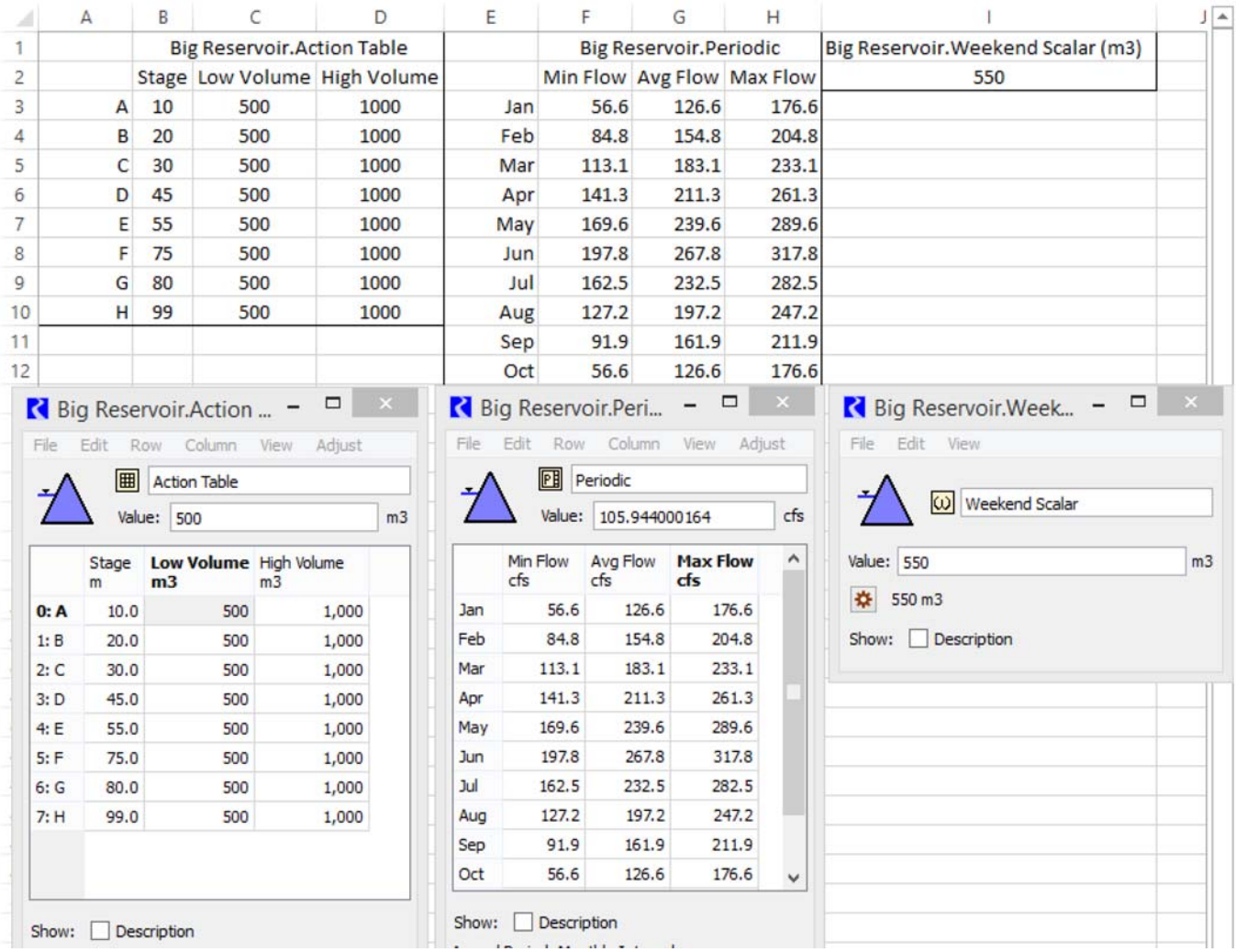


Table / Periodic / Scalar Slots Configuration Options

When the **Table / Periodic / Scalar Slots** option is chosen, you are not allowed to specify the format of the slots in the spreadsheet. Instead, there is a very specific format that is used. Figure 4.24 shows a sample workbook. Table slots have a header merged cell for the slot name, cells for column headers, and the columns that contain row headers and data columns. Periodic slots include a column of dates, one or more columns of data, and a header cell. Scalar slots are represented by a header cell and a value cell.

> **Note:** The associated slots in RiverWare are also shown. The sheet name is Run0 because that option (described below) is used.

Figure 4.24



		Big Reservoir.Action Table		
		Stage	Low Volume	High Volume
3	A	10	500	1000
4	B	20	500	1000
5	C	30	500	1000
6	D	45	500	1000
7	E	55	500	1000
8	F	75	500	1000
9	G	80	500	1000
10	H	99	500	1000

	Big Reservoir.Periodic		
	Min Flow	Avg Flow	Max Flow
Jan	56.6	126.6	176.6
Feb	84.8	154.8	204.8
Mar	113.1	183.1	233.1
Apr	141.3	211.3	261.3
May	169.6	239.6	289.6
Jun	197.8	267.8	317.8
Jul	162.5	232.5	282.5
Aug	127.2	197.2	247.2
Sep	91.9	161.9	211.9
Oct	56.6	126.6	176.6

Big Reservoir.Weekend Scalar (m3)
550

Table / Periodic / Scalar Slot DMI Limitations

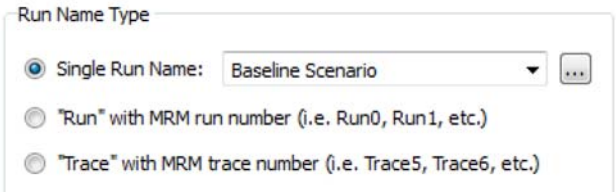
Functionality to import/export Table, Periodic, and Scalar slots from/to Excel has the following limitations:

- Multiple runs are supported by having sheets with either the RunN or TraceN format.
- Using an offset from the first row/column is not possible.
- The **Use Unit Name with Slot Name** is only supported for Scalar slots. Units in Table or Periodic columns are not supported.
- Column and row labels must be specified in Excel as strings, even when the labels are the string representation of numbers (which is common for row labels). Unfortunately, Excel tries to convert string numbers into Numbers. To prevent this, quote numeric column and row labels with a leading apostrophe, for example '10.
- Table or Periodic slots with the DATETIME unit type are not supported.

- Importing a Table slot from Excel can only resize the number of rows in RiverWare. The number of columns cannot be changed.
- Importing a Periodic slot from Excel must match exactly the number of rows and columns. Changing the number of column or rows is not allowed.
- Periodic slots export includes row labels (i.e. dates) in the Excel sheet. But on import, the dates must match the row labels or an error will occur. It is not possible to change the periodic dates, interval or period based on data in Excel. These must all be configured in the slot before importing values from Excel.

Run Name Type

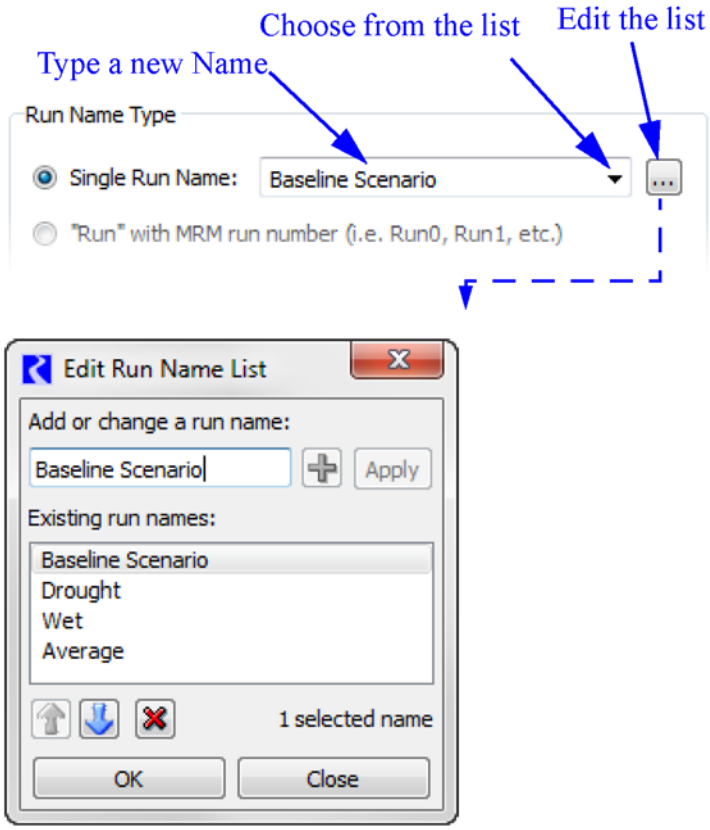
The **Run Name Type** options indicate how runs in the Excel workbook will be labeled.



Use one of the following options:

- **Single Run Name**. For a single run, the **Single Run Name** must be entered, which will be used to map to the appropriate dimension in Excel depending on the orientation selected (either row, column, or sheet). Either type the text directly into the field or use the chooser to select from a list of possible values. Use the button to edit the list of possible values. In the Edit Run Name List dialog, add, change, or delete possible Run Names using the

buttons. A Run Name List can be copied from one Excel Dataset to another using copy/paste operations in the right-click context menu.



- **"Run" with MRM run number**. For multiple runs, a single run name is not adequate to label each run individually. Two options are provided for labeling multiple runs. If **"Run" with MRM run number (i.e. Run0, Run1, etc.)** is selected, then the run name used for mapping to Excel will be the word Run followed by the index number of the run within the multiple run.

  Note: The index number used is zero-based to be compatible with how multiple run data has historically been written out of the ExcelWriter tool. The **Run** option cannot be used in the case of a distributed MRM run because run numbers start at zero in each distributed piece. A DMI error message is issued in this situation.

- "**Trace" with MRM trace number.** If **"Trace" with MRM trace number (i.e. Trace5, Trace6, etc.)** is selected, then the run name used for mapping to Excel will be the word Trace followed by the trace number of the run within the multiple run. Trace numbers are a one-based index into the runs, but the initial trace number for starting the multiple run can be specified by the user. Trace numbers can be used successfully with distributed MRM runs. See "Input DMI Runs" in *Solution Approaches* for details on setting up trace numbers in MRM.

  Note: Excel worksheet names (i.e. the worksheet tab label) must be less than or equal to 31 characters; this limit is imposed by Excel. If you choose to have Runs on sheets and you type in text in the **Single Run Name** field, it is limited to 31 characters. If you choose to have Slots as sheets and your slot name is more than 31 characters, when the DMI is run the slot is not written/read and a message is posted. In this case, use Name Maps to map the slot names to shorter names that will be accepted by Excel.

For Series Slots approach, an option is provided to **Use Begin Timestep Time Instead of End of Timestep**.



Timesteps in RiverWare are dated with the end of timestep time, that is, a 1 Day timestep for the last day of the year 2000 is dated 12/31/2000 24:00. If the begin timestep option is not checked, this date is written to (and is expected to be read from) Excel as 12/31/2000 23:59. A minute is subtracted compared to the RiverWare end of time so that the date appears in the correct month and year in Excel. (Excel has no concept of a 24:00 display, so 12/31/2000 24:00 is interpreted by Excel as 1/1/2001 00:00, which would appear to be the wrong month and year.)

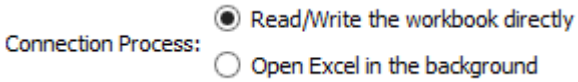
If **Use Begin Timestep Time Instead of End of Timestep** is checked, then the timestep is written to (and is expected to be read from) Excel as the beginning of timestep in RiverWare. For the above example, the time in Excel would be 12/31/2000 00:00. This option may be particularly useful if the user is entering data into Excel and is typing in 1 Day timesteps as 12/30/2000, 12/31/2000, 1/1/2001, etc., which end up by default with 00:00 hours and minutes in the Excel date format. With the begin timestep option checked, the data associated with these times will be moved into RiverWare for their correct timesteps.

An option is also provided with the header approach to **Use Unit Name with Slot Name**. In an output DMI, this option will attach the unit name to the slot name when the slot name is written out to its appropriate row, column, or sheet, depending on the orientation selected. The format is the slot name, a space, and the unit name in parentheses, such as Storage Reservoir.Inflow (cfs). In an input DMI, this option means that the DMI will expect the unit name to be attached to the slot name as indicated above when it is looking for the slot in the Excel headers or sheets.

Connection Process

The final part of the Excel DMI tab is the choice of Connection Process as shown in Figure 4.25.

Figure 4.25 Screenshot of the Excel Dataset Connection Process configuration



This configurations setting specified how the RiverWare DMI should connect or communicate with the Excel workbook. The two choices are:

- Read/Write the workbook directly: (Default) Read or write the data directly to the Excel workbook file, without opening the Excel application. This option typically has faster performance because Excel is not opened in the background. But, without opening Excel, there is no way to evaluate formulas in Excel or execute other automatic workbook operations.
- Open Excel in the background: Open Excel in the background, possibly to evaluate formulas, and then read or write the data. This was the original and only approach through RiverWare version 7.4. In this approach, the Excel application is opened in the background and then the DMI interacts with Excel through the Component Object Model (COM) interface. This approach requires that Excel is installed on the computer.

Name Mapping

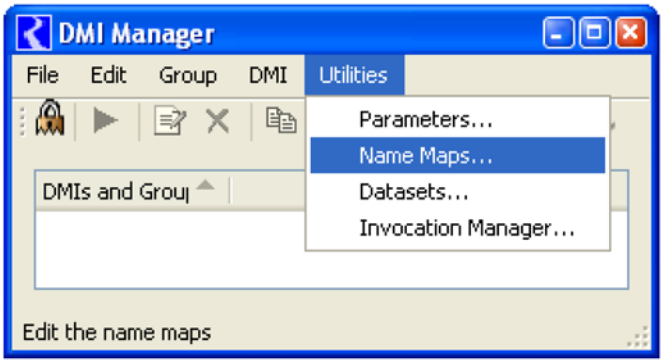
It is likely that RiverWare object or slot names do not match their associated database names. For example, the reservoir inflow is called *Inflow* in RiverWare while in DSS Part C, it is instead called *RES-INFLOW*. Because different databases use different naming conventions, there is no way for RiverWare to know the database's naming convention. The name mapping tool provides a tool for the user to specify the link between the two naming systems. The name mapping tool allows the user to define multiple name mappings.

Note: If the names are mapped within the database itself, as they are in HDB, a Name Map is not needed.

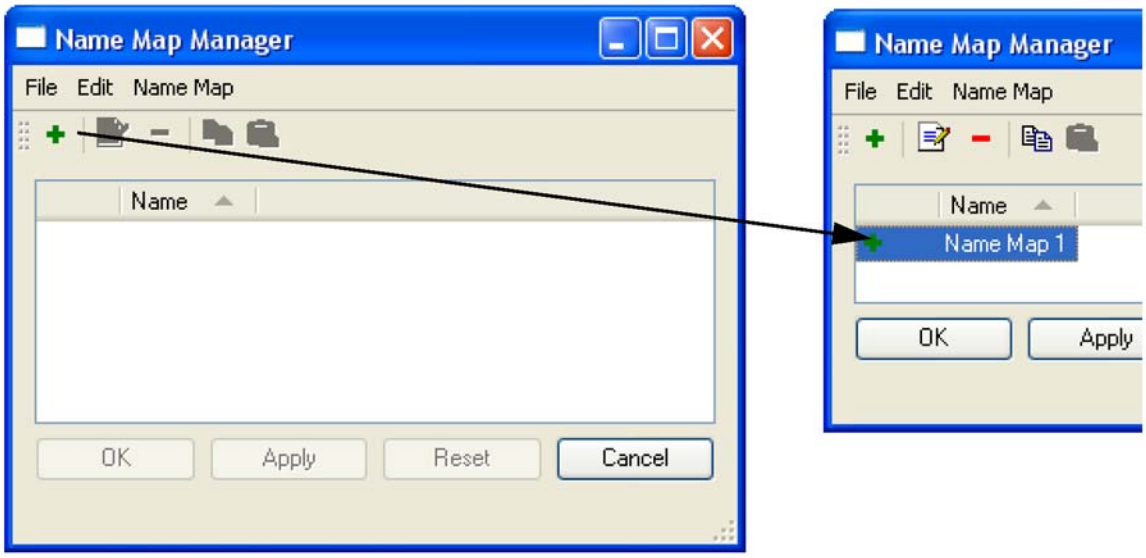
Note: For DSS datasets, there is no need to create a name map if the map is strictly to specify a different case. This is taken care of automatically. For example, a name map is not necessary for a reservoir in RiverWare called BigLake while in the DSS database, it is called BIGLAKE.

Configuring Name Mapping

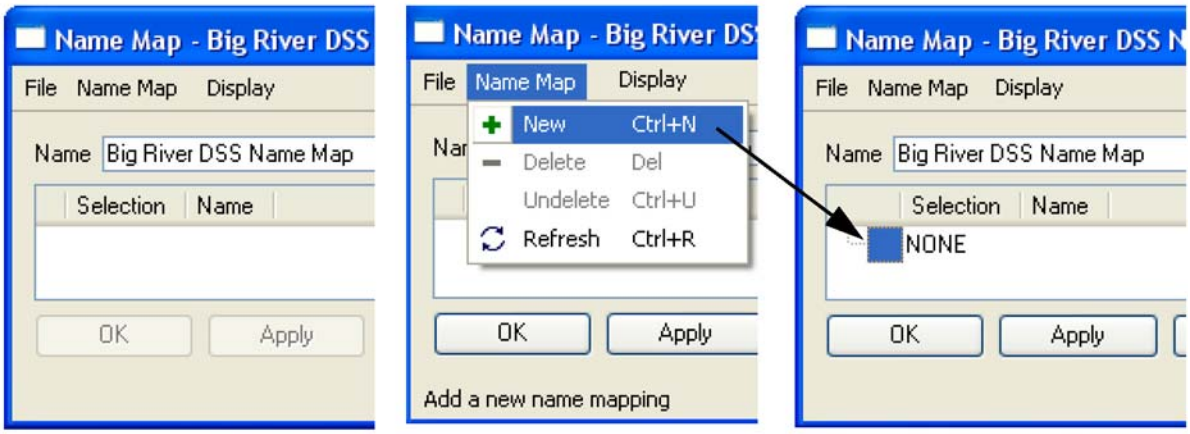
To bring up the name mapping editor from the DMI Manager, the user selects **Utilities**, then **Name Maps.**



This opens the Name Map Manager dialog. Selecting the + icon adds a new Name Map to the editor and gives it a default name.



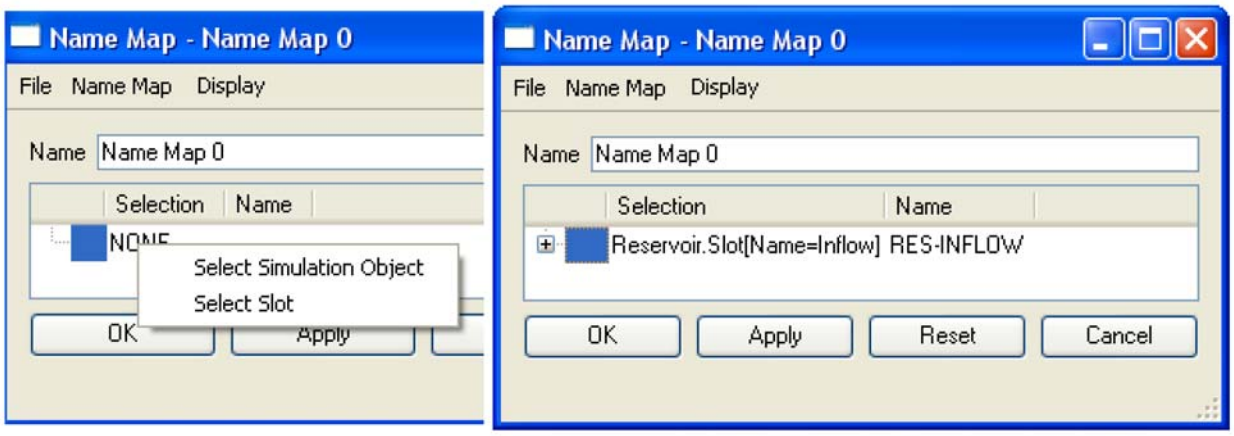
Double-clicking the newly added Name Map opens the Name Map dialog as shown in the following screenshot. In this dialog, the user is able to edit the name of the Name Map. Selecting **Name Map**, then **New** adds a new entry to the list.



Selecting the newly selected entry opens the context menu options shown in Figure 4.26. Either select **Select Simulation Object** or **Select Slot.** This action opens the selector dialog for the appropriate selection. In this dialog, the user selects the desired objects or slots that will be part of this Name Map. If accounting is enabled, supplies, accounts, exchanges, and paybacks are also available. Use the filters and wildcards to select generic slots. For example, the user could select Reservoir.Inflow as an entry. See "Creating a Name Map of All Reservoir.Inflow Slots" in *User Interface* for details on using the selector. The selections made here are considered dynamic root selections. That is, the selection is saved with the model file and is automatically updated if objects or slots are added to or removed from the model. This is especially useful when using wildcards.

To define the data name, in the Name Map manager, select the blank area to the right of the entry under the word Name. This is where the user specifies the database's variable name for the entry. Figure 4.26 shows an example.

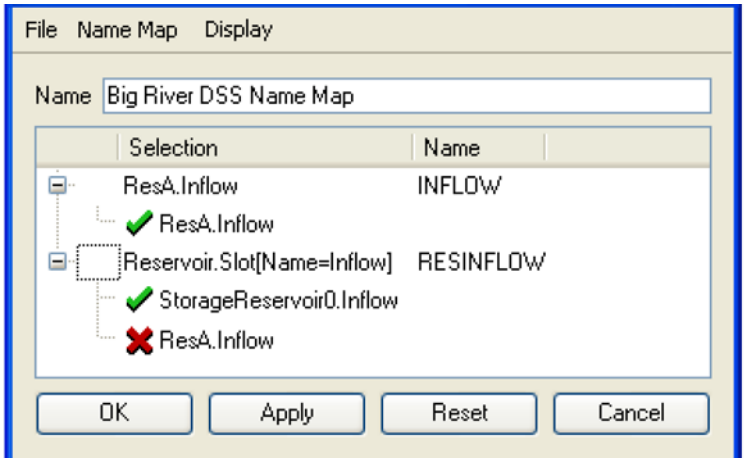
Figure 4.26

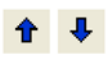


Name Map Ordering and Priorities

The entries in the Name Map are prioritized with the highest priority first. If there is a conflict between two entries, the entry with the highest priority takes precedence and the other entry becomes disabled with a red X. This means that if there is a red X, there is a higher priority selection that controls the item. For example, in Figure 4.27, the first entry for ResA.Inflow (INFLOW) takes priority over the second entry (RESINFLOW). The green check marks show that the first item is controlling the selection. This functionality allows users the functionality to create a name map in a generic case but redefine it in special case.

Figure 4.27



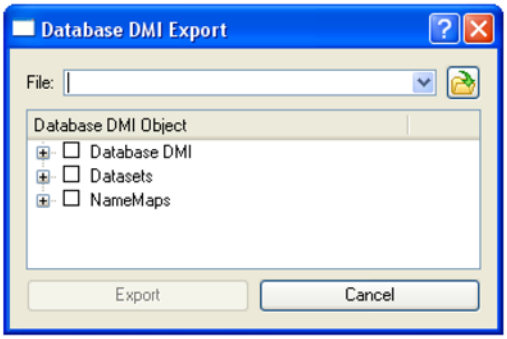
Because entries in the Name Map are prioritized, by default, they are ordered by this priority. The user is able to change the entries by using the **Display** menu and selecting **Custom**. Once in custom mode, the user select an entry (Selection and Name row) and then use the up and down arrows  or **Display**, then **Move Up** or **Display**, then **Move Down** file choices to rearrange rows. To return to prioritized mode, the user selects **Display**, then **Priority**. Regardless of the ordering, the name maps are stored according to priority; custom order is just for display.

Sharing Database DMIs

The Database DMI utility includes a feature that allows the user to Export and Import the Database DMI configurations. The user is able to export and import Database DMIs, Datasets, and Name Maps. Following are instructions and descriptions of both exporting and importing Database DMI configurations.

Export

To export the Database DMI configuration, the user selects **File**, then **Export Database DMI** from the DMI Manager. The Database DMI Export dialog appears.

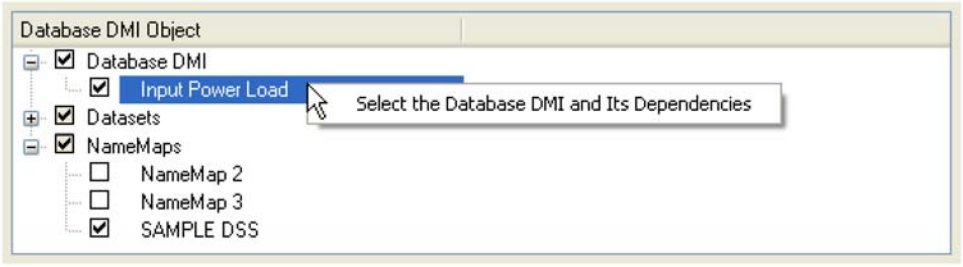


The user types a file path into the **File** field or selects an existing file using the browser button.



The user selects the ⊞ symbol to expand the tree view. Selecting the check boxes selects the Database DMI, Dataset or NameMap to export. Selecting the top tree level multiple times cycles through selecting all of the items, selecting none of the item, and the last selection made. Right-clicking on a Database DMI or a Dataset name, a context menu allows the user to **Select the Database DMI and Its Dependencies** as shown in Figure 4.28.This allows the user to select not only the Database DMI, but also the Dataset and/or NameMap associated with the selection.

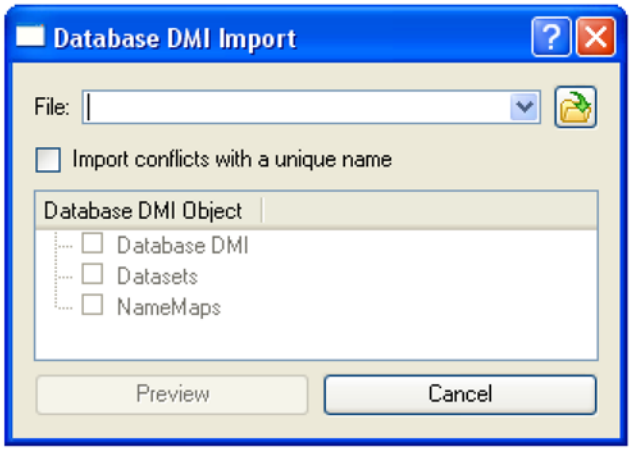
Figure 4.28



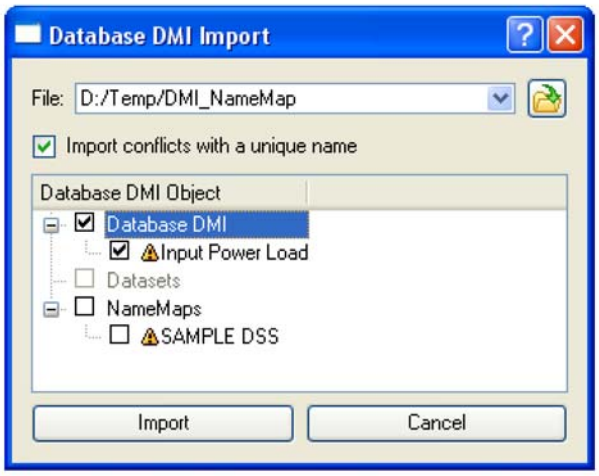
Once a file is entered and a selection is made, the **Export** button becomes active. Selecting **Export** will export the configuration to the file.

Import

Importing a set of configurations is similar to exporting. The user selects the **File**, then **Import Database DMI** menu on the DMI Editor to open the Database DMI Import dialog.



The user types in the file name or uses the browser to select the file to import. Selecting the **Preview** button reads the file and displays the items in the file. Selecting the  symbol expands the tree view.



The user then selects the items to import. If the item already exists in the model, a small warning sign is displayed. Toggle on the **Import conflicts with a unique name** button to import the item and give it a new name. In addition, when this option is checked the imported Database DMIs will be modified access any newly imported Datasets and Name Maps that are also imported.

If the checkbox is cleared and there is a conflict, the import will overwrite any existing items of the same name in the model. When the user is satisfied with the selection, selecting **Import** brings the items into the model.

Chapter 4
Database DMI