



**Technical Documentation Version 6.2**

---

# **Data Management Interface**

---



**C A D S W E S**

**Center for Advanced Decision Support for Water and Environmental Systems**

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.

# Data Management Interface Table of Contents

<b>Introduction .....</b>	<b>1</b>
<b>DMI User Interface .....</b>	<b>1</b>
<b>Accessing DMI's and the DMI Manager .....</b>	<b>2</b>
<b>DMI Manager Dialog .....</b>	<b>2</b>
Conventions .....	3
Creating and Deleting DMIs .....	4
Opening the DMI Editor Dialog .....	4
Creating and Deleting DMI Groups.....	4
Top-Level DMIs and Member DMIs .....	5
Editing DMI Groups.....	5
Sorting DMIs and DMI groups.....	7
Opening the DMI Parameter Dialog .....	7
Locking the DMI Manager Dialog.....	7
<b>DMI Parameter Dialog .....</b>	<b>8</b>
Creating and Deleting Parameters .....	8
Editing a Parameter .....	9
Editing a Parameter's Name.....	9
Editing a Parameter's Type and Attributes .....	10
Exporting and Importing Parameters.....	12
Reconciling Parameter Edits .....	13
<b>DMI Invocation Manager Dialog .....</b>	<b>15</b>
Overview .....	15
Views .....	15
Clearing Values .....	16
Deleting Invocation Records.....	17
<b>Invoking DMIs and DMI Groups .....</b>	<b>17</b>
Warning and Error Messages .....	19
DMI Status .....	20
<b>Icon Quick Reference .....</b>	<b>20</b>
<b>Control File-Executable Approach .....</b>	<b>23</b>
Introduction .....	23
Control File .....	24
Object.Slot Specifications.....	25
File Name and Directory specification.....	26
Data Files .....	30
DMI executable .....	33

---

<b>Meta-control File .....</b>	<b>33</b>
<b>DMI Editor Dialog .....</b>	<b>34</b>
<b>Name .....</b>	<b>34</b>
<b>Type .....</b>	<b>34</b>
<b>Confirm Warnings Toggle.....</b>	<b>34</b>
<b>Record Invocations .....</b>	<b>35</b>
<b>Allow Spaces in File Paths.....</b>	<b>35</b>
<b>Line Ending .....</b>	<b>35</b>
<b>Control File .....</b>	<b>35</b>
<b>Executable .....</b>	<b>36</b>
<b>User Parameters.....</b>	<b>36</b>
<b>Command Line .....</b>	<b>37</b>
<b>Database DMI .....</b>	<b>39</b>
<b>Overview .....</b>	<b>39</b>
<b>Creating a Database DMI .....</b>	<b>41</b>
<b>Name and Type.....</b>	<b>43</b>
<b>DMI Configuration.....</b>	<b>44</b>
<b>DSS Dataset General Configuration.....</b>	<b>50</b>
<b>DMI User Parameters .....</b>	<b>50</b>
<b>Examples .....</b>	<b>51</b>
<b>Datasets .....</b>	<b>59</b>
<b>DSS Datasets.....</b>	<b>59</b>
<b>HDB Datasets .....</b>	<b>62</b>
<b>Excel Datasets.....</b>	<b>70</b>
<b>Name Mapping .....</b>	<b>81</b>
<b>Configuring Name Mapping .....</b>	<b>81</b>
<b>Re-ordering entries in the Name Map .....</b>	<b>83</b>
<b>Sharing Database DMI's .....</b>	<b>84</b>
<b>Export.....</b>	<b>84</b>
<b>Import.....</b>	<b>85</b>

# Introduction

---

## 1. Introduction

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

There are two types of DMI's:

- **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. Click [HERE \(Section 3\)](#) to go directly to this section.
- **Database DMI:** Currently, RiverWare supports a direct connection between three databases: the Hydrologic Engineering Center's Data Storage System (HEC-DSS), the Hydrologic DataBase (HDB), and Microsoft Office Excel. In this configuration, all of the slot specification is done within RiverWare using Datasets, Name Maps, and Database DMIs. Click [HERE \(Section 4\)](#) to go directly to this section.

In this document, we describe the DMI user interface components that are general to both approaches. Then we describe the Control File-Executable approach and its specific user interface components. Finally, we describe the Database DMI approach and its specific user interface components.

---

## 2. DMI User Interface

This section describes the user interface used to execute and create DMI's. This section applies to both the Control File-Executable and Database DMI approach. Additional user interface for the Database DMI section is presented [HERE \(Section 4\)](#).

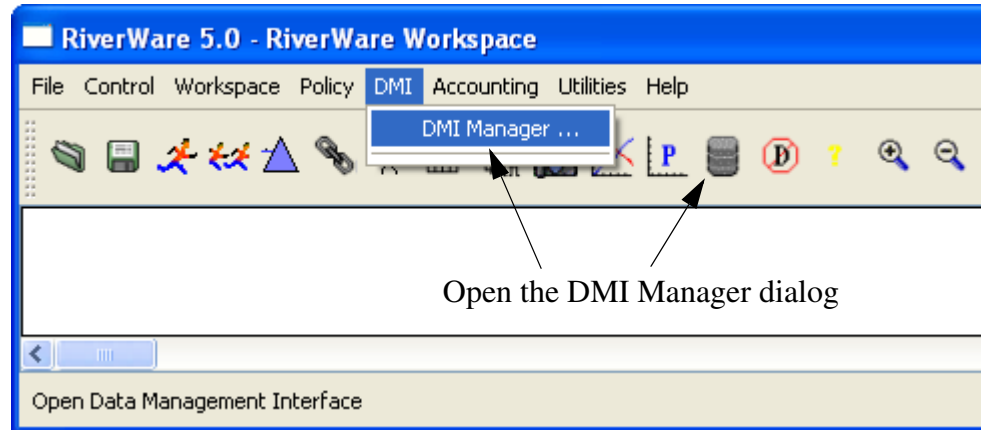
## 2.1 Accessing DMI's and the DMI Manager

The Workspace provides a DMI menu and a DMI

toolbar button.



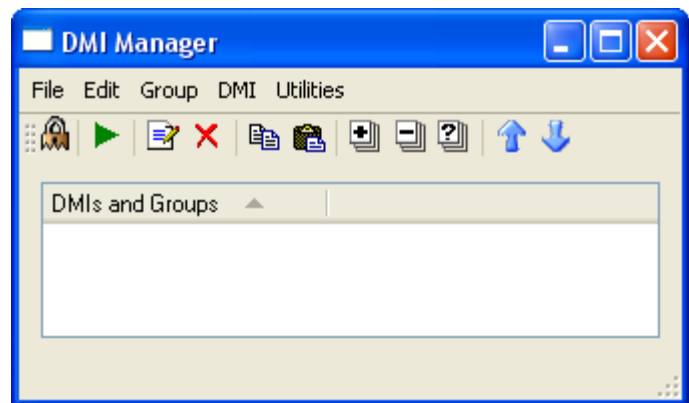
Selecting either the **DMI** → **DMI Manager** menu item or the toolbar button opens the DMI Manager dialog. Once one or more DMIs are defined, the DMI menu also contains these DMIs which can be invoked directly from the workspace, discussed further [HERE \(Section 2.5\)](#).



## 2.2 DMI Manager Dialog

The DMI Manager dialog, shown below, enables the user to:

- 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

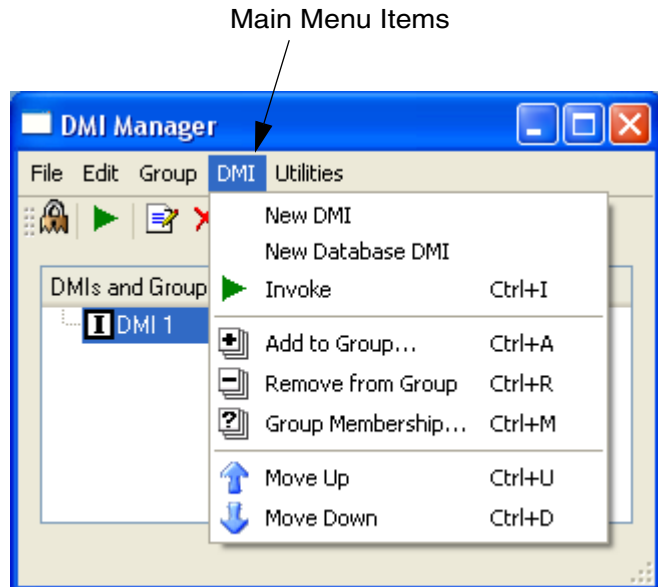


## 2.2.1 Conventions

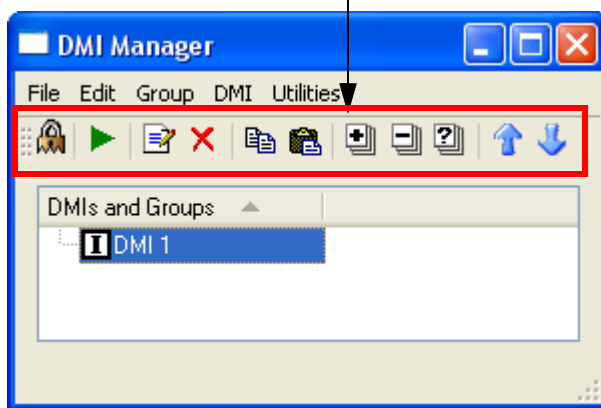
In the DMI Manager and related dialogs, actions can be performed using one or more of the following, each of which acts on the current selection:

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

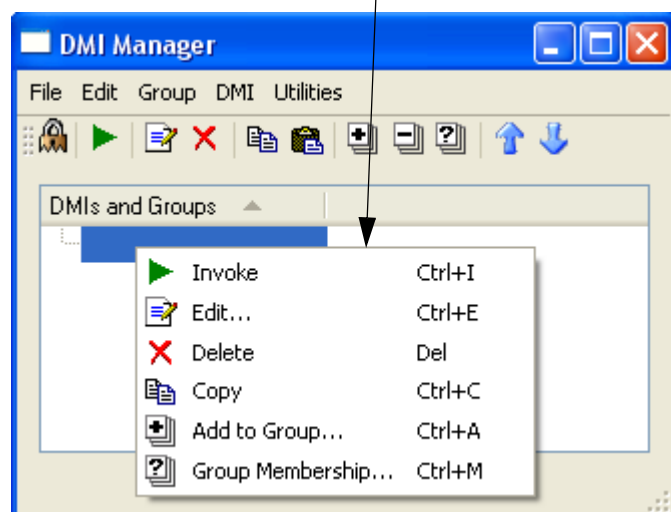
Main menu items, toolbar buttons and context menu items performing the same action share the same icon. As an example, from the DMI Manager dialog a user can invoke the selected DMI identically using the **DMI** → **Invoke** main menu item, the **Invoke** toolbar button, the DMI **Invoke** context menu item and the **Control-I** keyboard accelerator:



Toolbar buttons



Context Menu Items (right click)



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.

Also, the document refers to “clicking twice” in dialog list cells to enable editing. The paradigm is really “click pause click” where:

- The first click in a row selects the row.
- After a brief pause, all subsequent clicks in the selected row enable editing in the clicked-in cell (and disable editing in the previously clicked-in cell). (In this context “brief” means long enough to not be considered a double-click.)
- The first click in a new row disables editing in the previously clicked-in row and selects the new row.

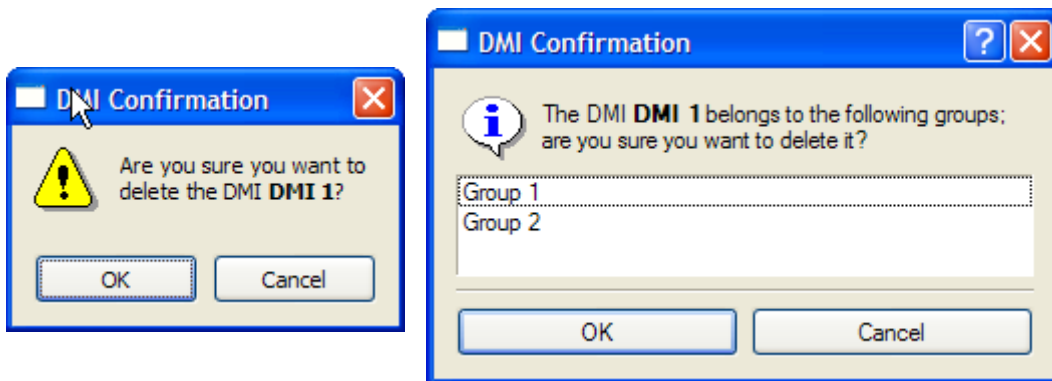
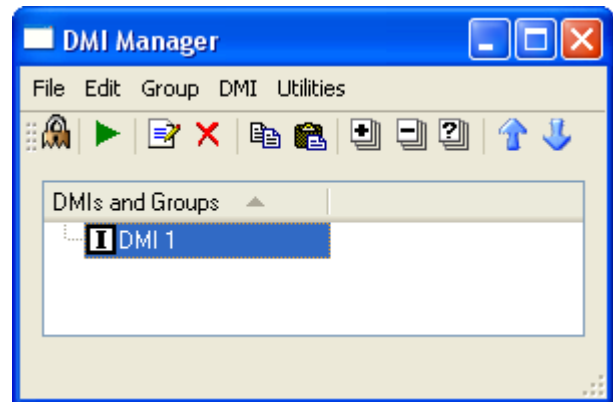
## 2.2.2 Creating and Deleting DMIs

The user creates a new DMI with the **DMI → New** main menu item; the DMI has a default name:

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

The user deletes a DMI with the **Edit → Delete** main menu item; the user will be asked to confirm the deletion:

Note there are two confirmation dialogs - the dialog on the left is shown when the DMI is not a member of any DMI groups; the dialog on the right is shown when the DMI is a member of one or more DMI groups. (DMI groups are discussed [HERE \(Section 2.2.4\)](#).)



## 2.2.3 Opening the DMI Editor Dialog

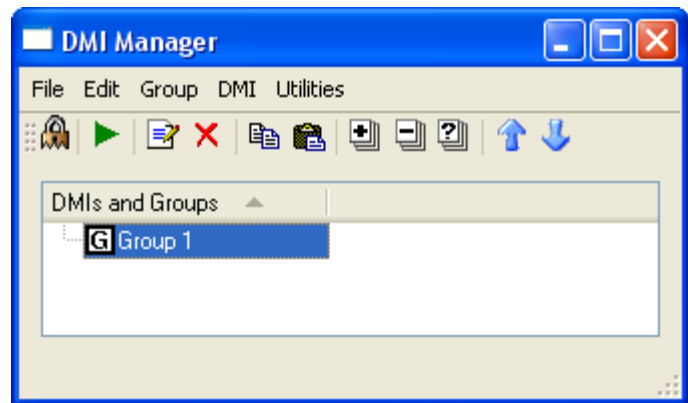
The user opens the DMI Editor dialog with the **Edit → Edit** main menu item. The DMI manager is different for the two types of DMI's. The DMI Editor dialog for the Control File-Executable approach is discussed [HERE \(Section 3.5\)](#). The DMI Editor for the Database DMI is discussed [HERE \(Section 4.2\)](#).

## 2.2.4 Creating and Deleting DMI Groups

A DMI group is an ordered collection of zero or more DMIs (although groups of zero or one DMIs aren't very interesting). When a DMI group is invoked its DMIs are invoked in order, until all have completed or one has failed.

The user creates a new DMI groups with the **Group** → **New** main menu item; the DMI group has a default name:

**G** indicates a DMI group.

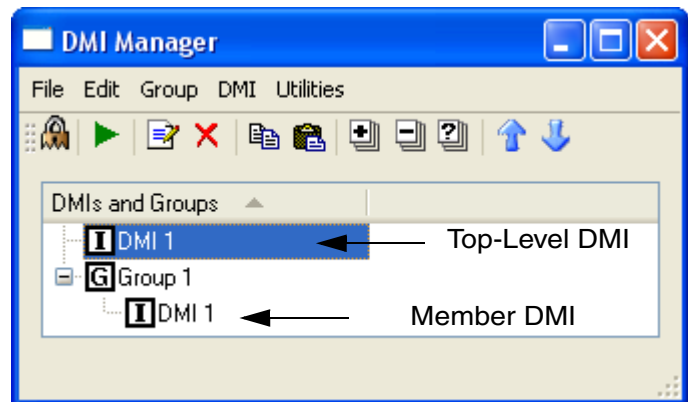


The user deletes a DMI group with the **Edit** → **Delete** main menu item; the user will be asked to confirm the deletion:



### 2.2.5 Top-Level DMIs and Member DMIs

Now that DMIs and groups have been discussed, it's worth introducing the concept of a top-level DMI. 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:



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

### 2.2.6 Editing DMI Groups

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)

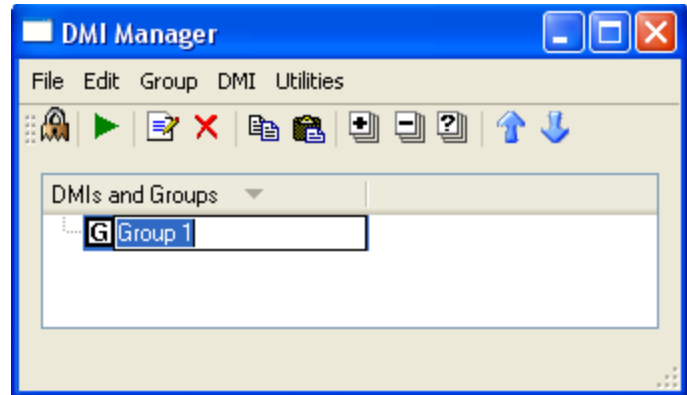
- Querying a DMI's group membership

To edit a DMI group's name the user clicks twice in the name cell, to edit the name in place:

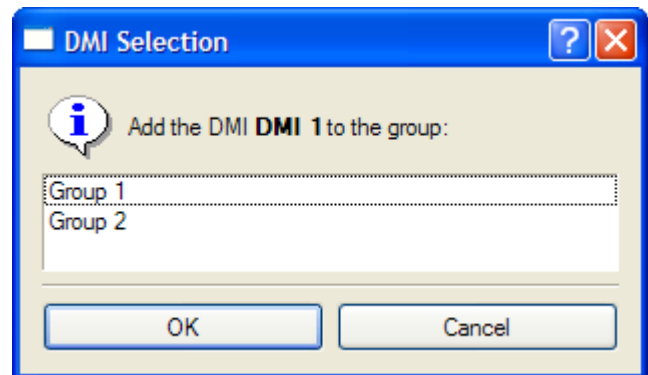
After editing the name it's important to hit **Enter**, to ensure the edit is accepted. The possible errors are:

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

To add a top-level DMI to a group the user selects the **DMI → Add to Group...** main menu item. The user is presented with a dialog listing the available groups; the user selects the group and clicks **OK**:



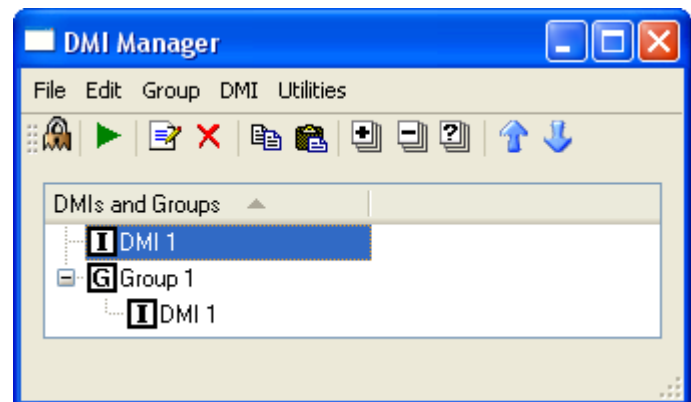
If no groups are available (e.g. there are no groups or the DMI is a member of all groups) the user will be notified. After successful completion the DMI is shown as a member of the group:



To remove a member DMI from a group the user selects the **DMI → Remove from Group** main menu item.

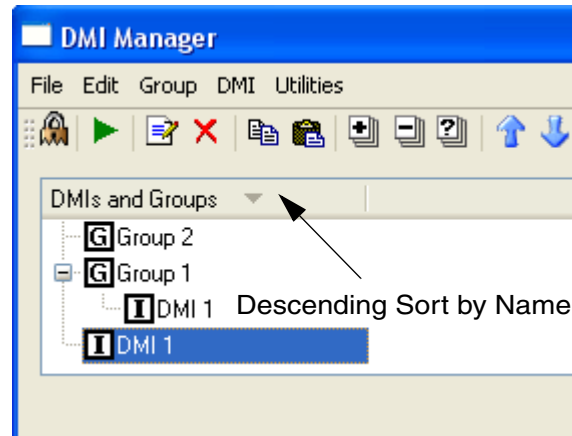
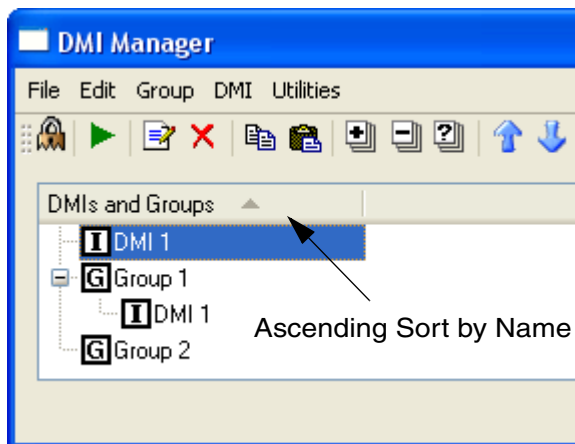
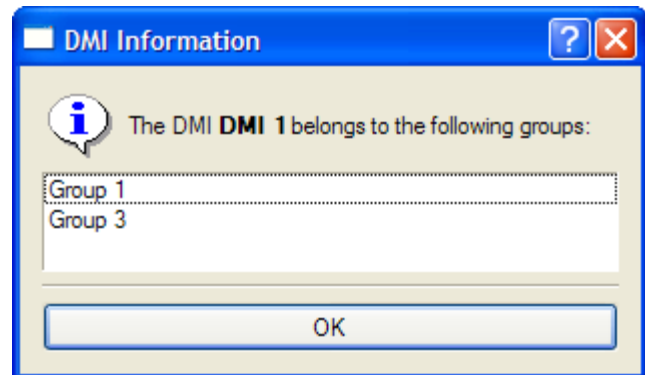
To move a member DMI within a group the user selects either the **DMI → Move Up** main menu item or the **DMI → Move Down** main menu item.

To query a DMI's group membership the user selects the **DMI → Group Membership** main menu item; the DMI's group membership is displayed:



### 2.2.7 Sorting DMIs and DMI groups

Currently sorting of DMIs and DMI groups is limited to sorting by name, in either ascending or descending order. The user toggles between orders by clicking the **DMIs and Groups** column header.



### 2.2.8 Opening the DMI Parameter Dialog

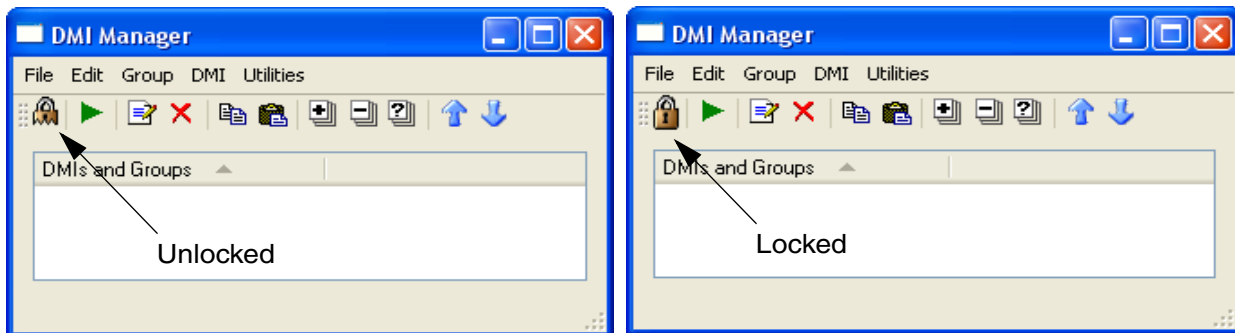
The user opens the DMI Parameter dialog from the DMI manager using the **Utilities** ➔ **Parameters** menu item. (The DMI Parameter dialog is discussed in detail below.)

### 2.2.9 Locking the DMI Manager Dialog

To lock the DMI manager dialog the user selects the **Edit** ➔ **Lock** main menu item. Before locking the dialog the user must close the DMI parameter dialog and the DMI editor dialogs (and will be warned if the dialogs are open).

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

To unlock the DMI Manager dialog the user selects the **Edit** ➔ **Unlock** main menu item.



## 2.3 DMI Parameter Dialog

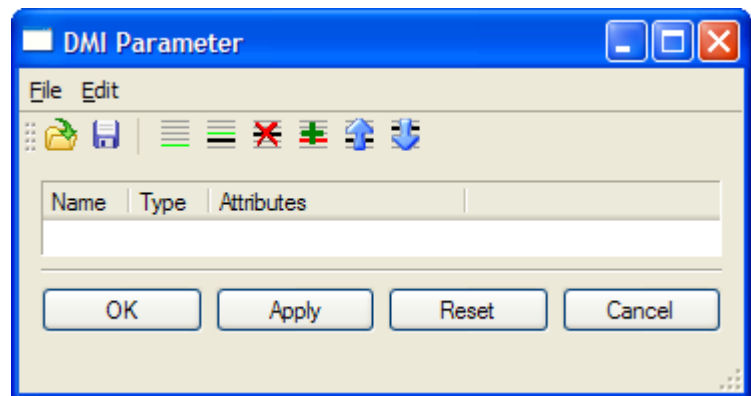
The DMI Parameter dialog, shown below, enables the user to:

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

### 2.3.1 Creating and Deleting Parameters

To create a parameter the user either:

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

To undelete a row the user selects the **Edit** → **Undelete Row** main menu item; the **-** mark is removed from the parameter.

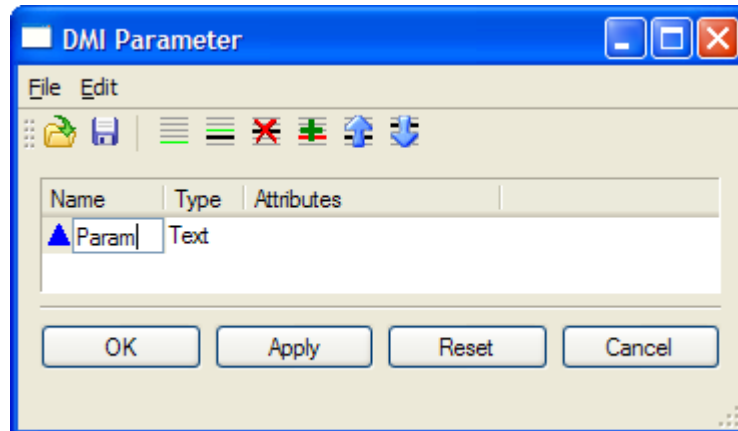
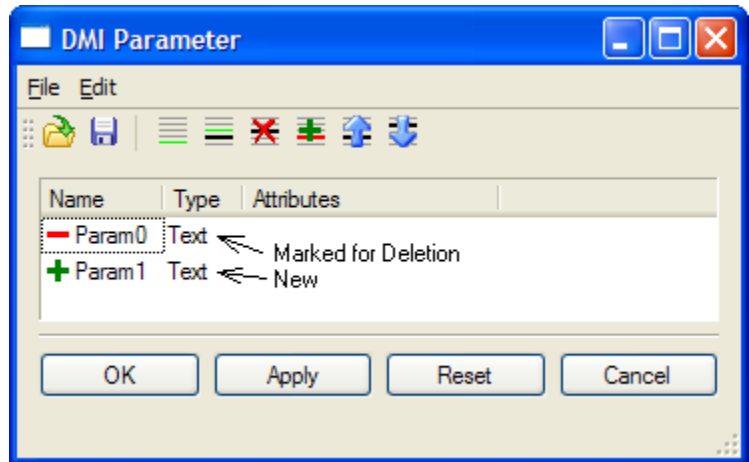
### 2.3.2 Editing a Parameter

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.

### 2.3.3 Editing a Parameter's Name

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

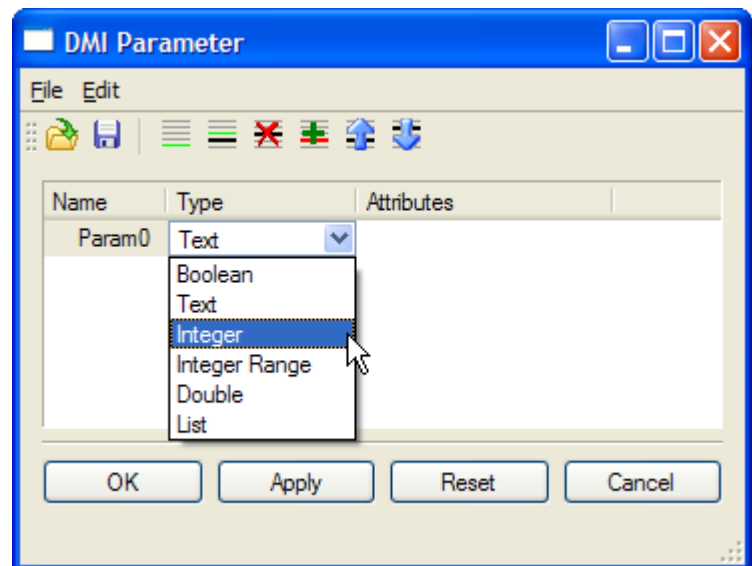
- A blank name
- A name containing invalid characters (characters other than letters, digit and underscore)
- A duplicate name (the same name as another parameter)



### 2.3.4 Editing a Parameter's 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's type, the user click-pause-clicks in the **Type** cell, to activate the type menu and then selects the parameter type:

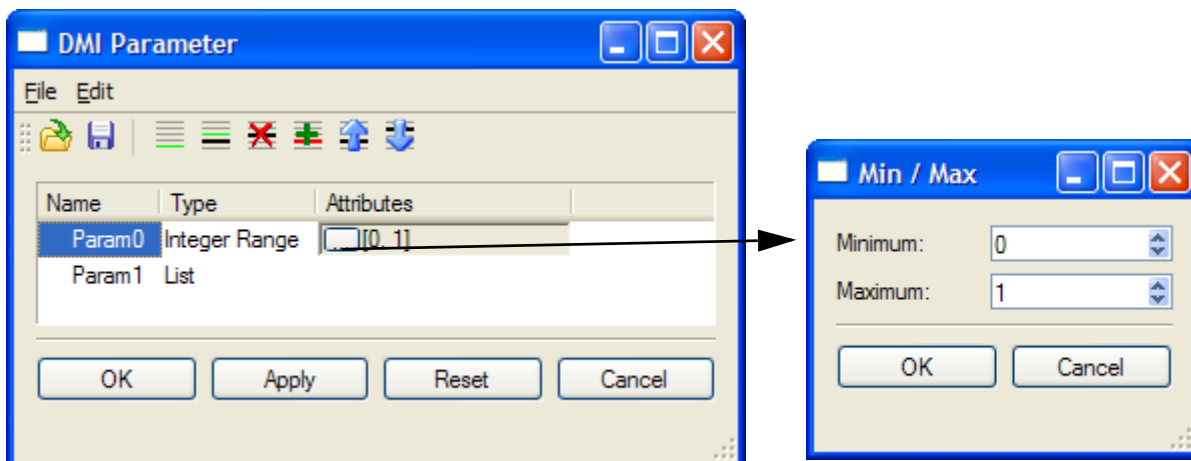
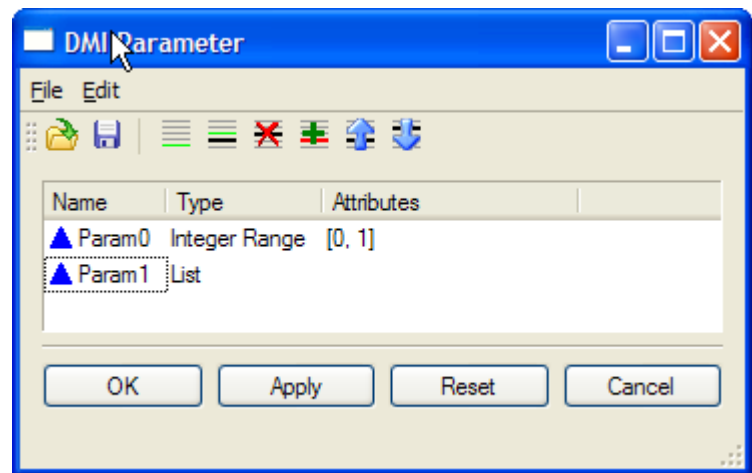


If the selected type has attributes the default attributes are displayed in the **Attributes** column:

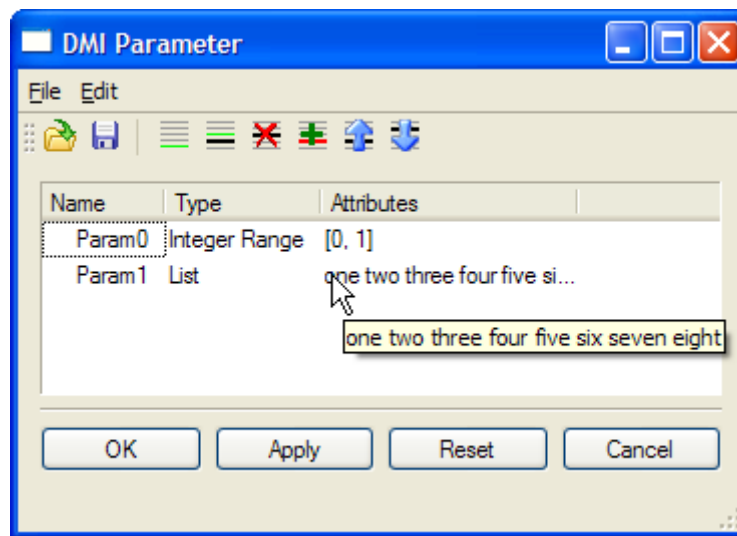
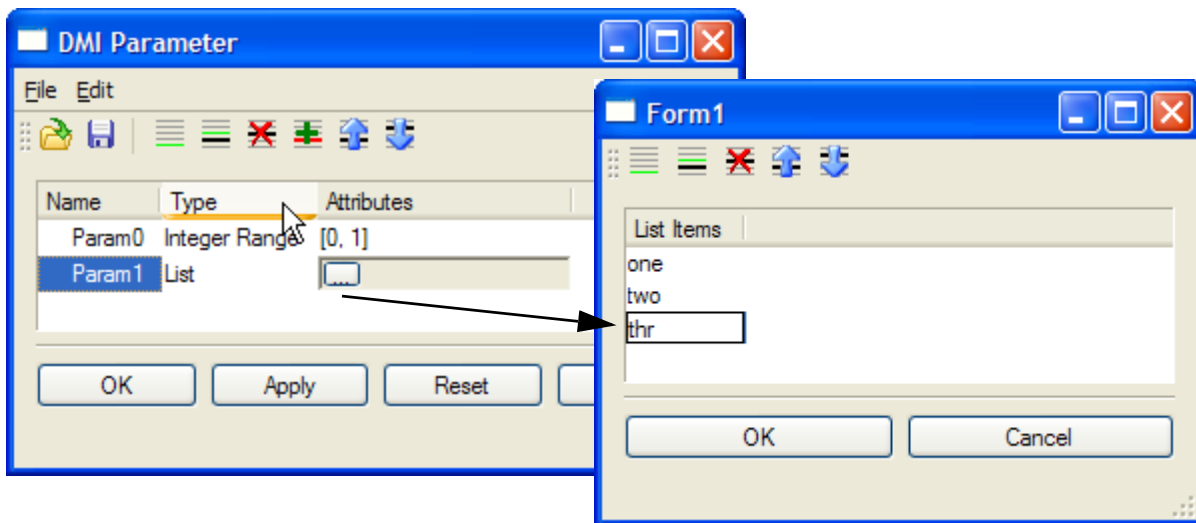
Note that a **List** parameter's default attribute is an empty list, so nothing is displayed.

To edit a parameter's attributes the user clicks twice in 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:



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). If the resulting list contains more text than will fit in the **Attributes** cell, “...” is displayed and the entire list is displayed in a tooltip when the mouse pointer is paused on the **Attributes** cell (second image).



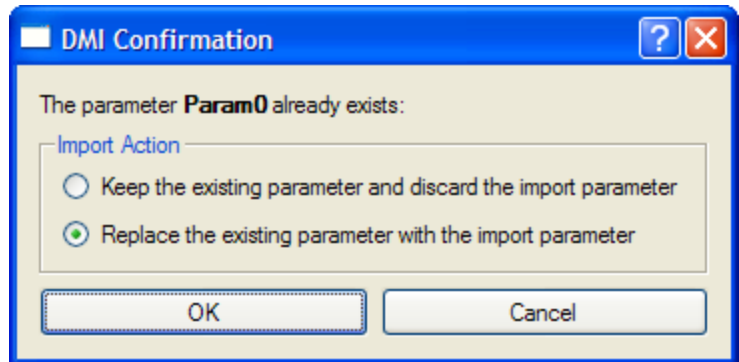
### 2.3.5 Exporting and Importing Parameters

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

To export parameters the user selects the **File** → **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, which is necessary for sharing the parameters between Solaris and Windows.

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

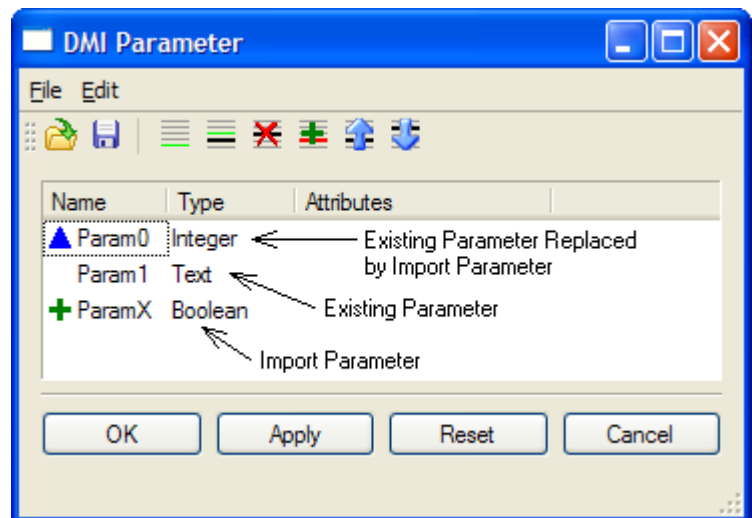
Parameters are imported by name; if the file contains a parameter which already exists, the user is asked whether to keep the existing parameter or replace it with the import parameter:



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



### 2.3.6 Reconciling Parameter Edits

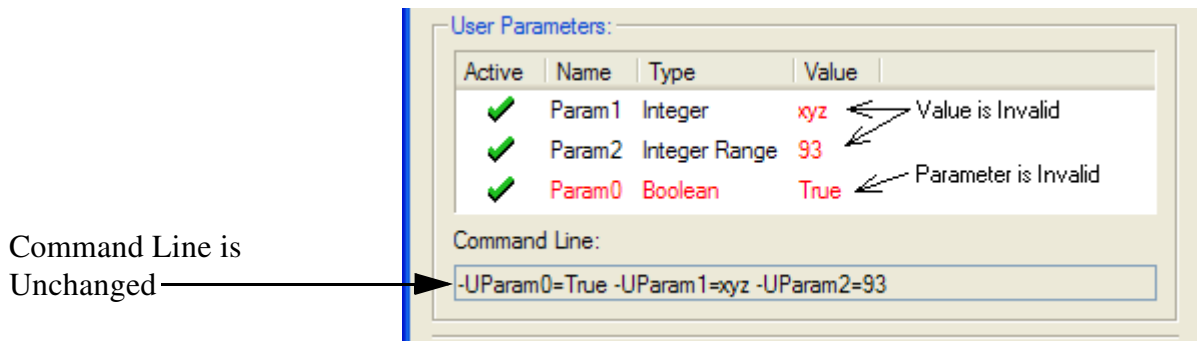
When the DMI Parameter dialog edits parameters it can invalidate DMIs using the parameters. Consider **Param0** (a **Boolean** parameter), **Param1** (a **Text** parameter) and **Param2** (an **Integer Range** parameter):

- 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:



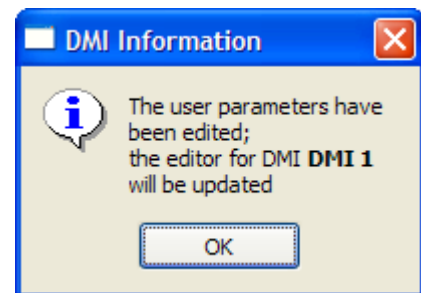
**Param0** is shown entirely in red to indicate it is deleted; the user:

- 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, the user is notified that the DMI Editor dialog is being updated with the message shown in the screenshot:

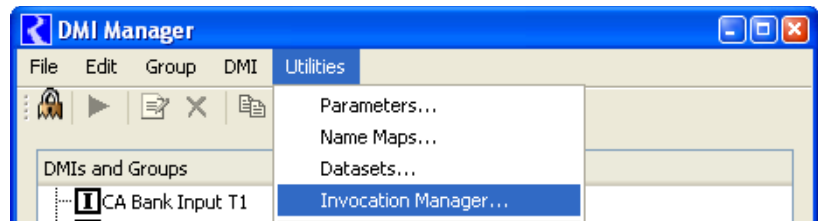


## 2.4 DMI Invocation Manager Dialog

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 DMI's that have been explicitly configured to do so. Information on this can be found: [HERE \(Section 3.5.4\)](#) for Control File - Executable DMI's or [HERE \(Section 4.2\)](#) for Database DMI's.

The dialog is opened from the **DMI Manager** dialog using the **Utilities** → **Invocation Manager** menu.



### 2.4.1 Overview

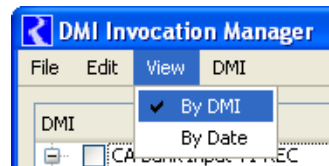
For each invocation the dialog displays:

- The DMI's 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.

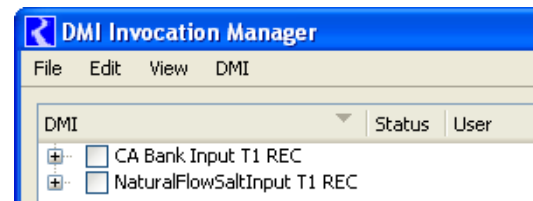
The **DMI Invocation Manager** is a singleton; that is, no other edits or views to any RiverWare window can be made while the **DMI Invocation Manager** is open. It must be closed first. Use the **File** → **Close Window** menu, Ctrl-W shortcut or **Close** button.

### 2.4.2 Views

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

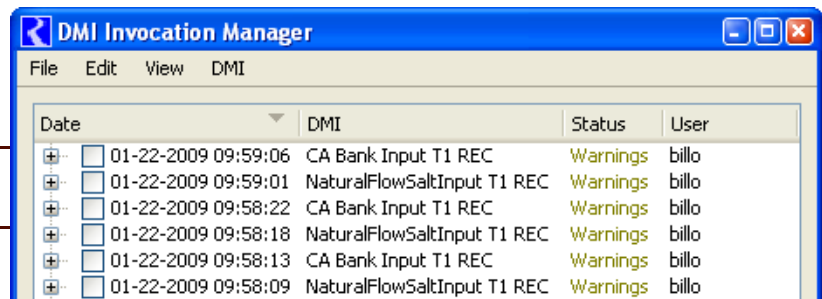


In the **By DMI** view, the invocations are grouped by DMI, with the invocations in reverse-chronological order (most recent is on the top) as shown in the screenshot.



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

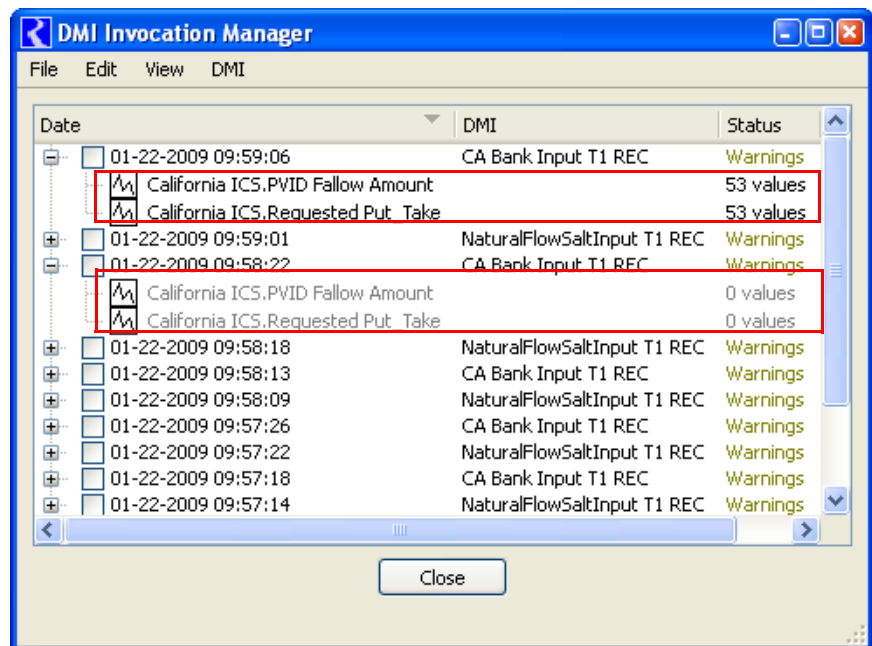
**Note:** Note, the columns are sortable by clicking on the column heading.



Date	DMI	Status	User
01-22-2009 09:59:06	CA Bank Input T1 REC	Warnings	billio
01-22-2009 09:59:01	NaturalFlowSaltInput T1 REC	Warnings	billio
01-22-2009 09:58:22	CA Bank Input T1 REC	Warnings	billio
01-22-2009 09:58:18	NaturalFlowSaltInput T1 REC	Warnings	billio
01-22-2009 09:58:13	CA Bank Input T1 REC	Warnings	billio
01-22-2009 09:58:09	NaturalFlowSaltInput T1 REC	Warnings	billio

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 the example shown, 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.



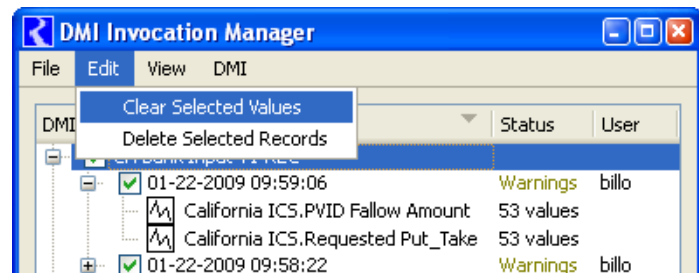
Date	DMI	Status
01-22-2009 09:59:06	CA Bank Input T1 REC	Warnings
	California ICS.PVID Follow Amount	53 values
	California ICS.Requested Put_Take	53 values
01-22-2009 09:59:01	NaturalFlowSaltInput T1 REC	Warnings
01-22-2009 09:58:22	CA Bank Input T1 REC	Warnings
	California ICS.PVID Follow Amount	0 values
	California ICS.Requested Put_Take	0 values
01-22-2009 09:58:18	NaturalFlowSaltInput T1 REC	Warnings
01-22-2009 09:58:13	CA Bank Input T1 REC	Warnings
01-22-2009 09:58:09	NaturalFlowSaltInput T1 REC	Warnings
01-22-2009 09:57:26	CA Bank Input T1 REC	Warnings
01-22-2009 09:57:22	NaturalFlowSaltInput T1 REC	Warnings
01-22-2009 09:57:18	CA Bank Input T1 REC	Warnings
01-22-2009 09:57:14	NaturalFlowSaltInput T1 REC	Warnings

### 2.4.3 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** → **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.

The following are some limitations on the clearing of DMI values:



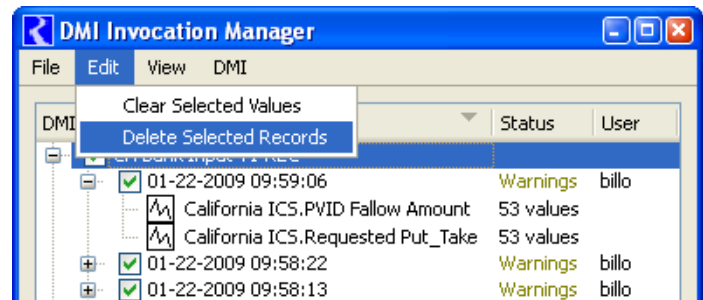
Date	DMI	Status	User
01-22-2009 09:59:06	CA Bank Input T1 REC	Warnings	billio
	California ICS.PVID Follow Amount	53 values	
	California ICS.Requested Put_Take	53 values	
01-22-2009 09:58:22	CA Bank Input T1 REC	Warnings	billio

- 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.
- The DMI\_INPUT flag will be introduced; values set by a DMI will no longer have the INPUT “I” flag - they will instead have the DMI\_INPUT “Z” flag.

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

#### 2.4.4 Deleting Invocation Records

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

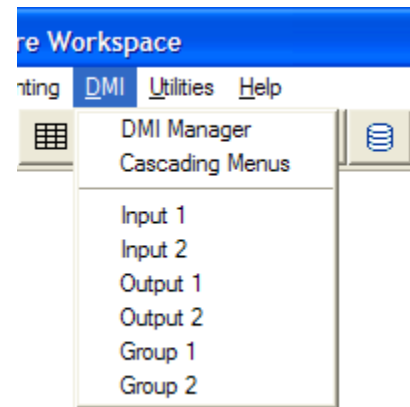


#### 2.5 Invoking DMIs and DMI Groups

There are 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**.

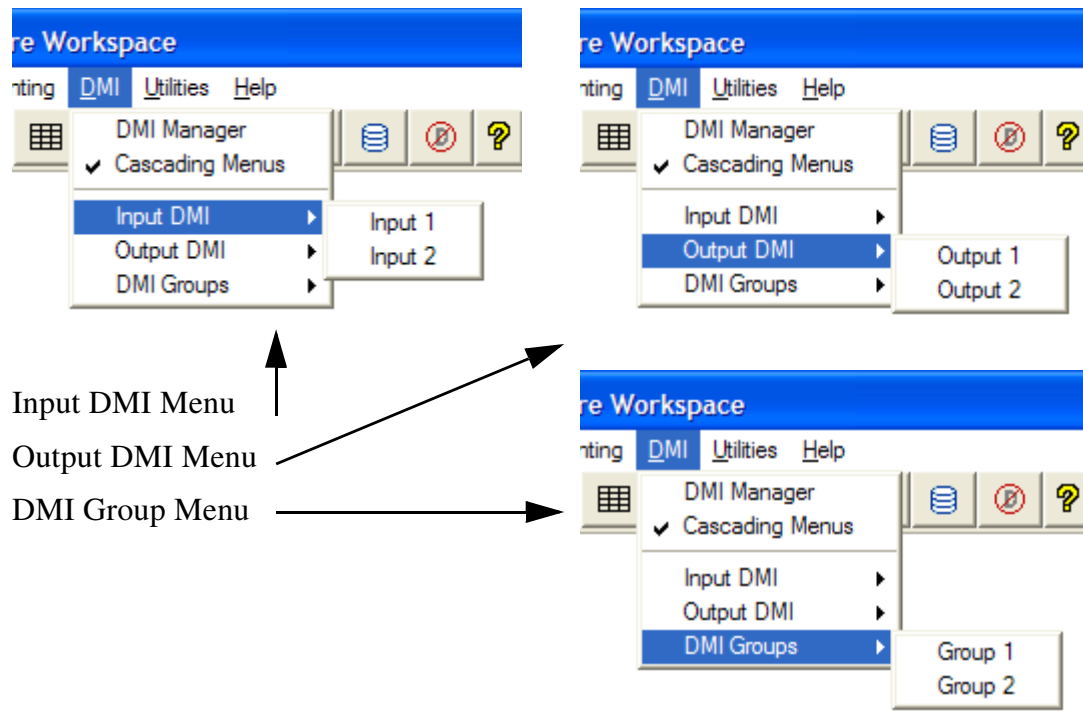
**DMI Manager:** To invoke a top-level DMI the user selects the **DMI** → **Invoke** main menu item; to invoke a DMI group the user selects the **Group** → **Invoke** main menu item.

**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 toggle between a conventional menu (right) and a cascading menu (below). In the conventional menu the DMIs and DMI groups are sorted by name within type (first input DMIs then output DMIs then DMI groups):



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

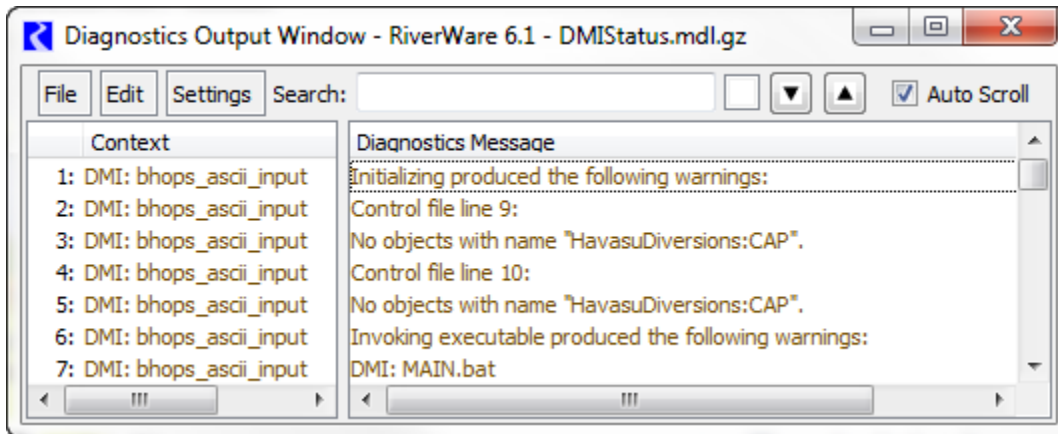
**DMI Invocation Manager:** The **DMI Invocation Manager** has a similar menu to the Workspace **DMI** menu. The only difference is that there is no **DMI Manager** item on the **DMI Invocation Manager** → **DMI** menu. (The **DMI Invocation Manager** can only be accessed from the **DMI Manager** so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



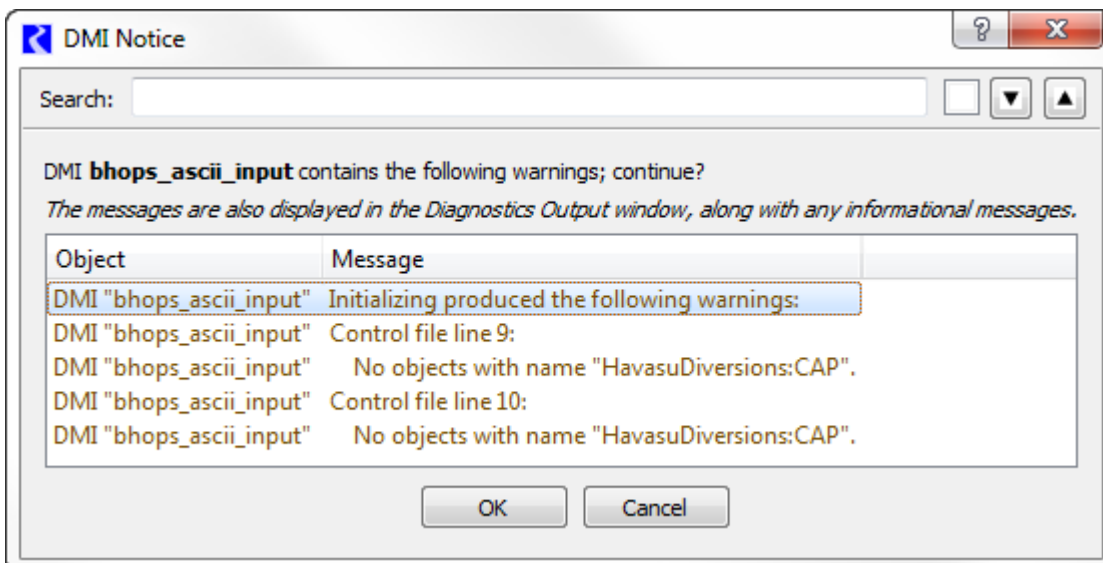
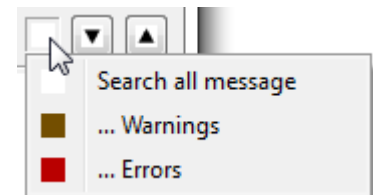
**RPL Blocks:** A DMI or a DMI group can be executed from a RPL block (typically a rule) as either a pre-execution or post-execution. This aspects of DMIs is described [HERE \(RPLUserInterface.pdf, Section 1.6.4\)](#).

## 2.5.1 Warning and Error Messages

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

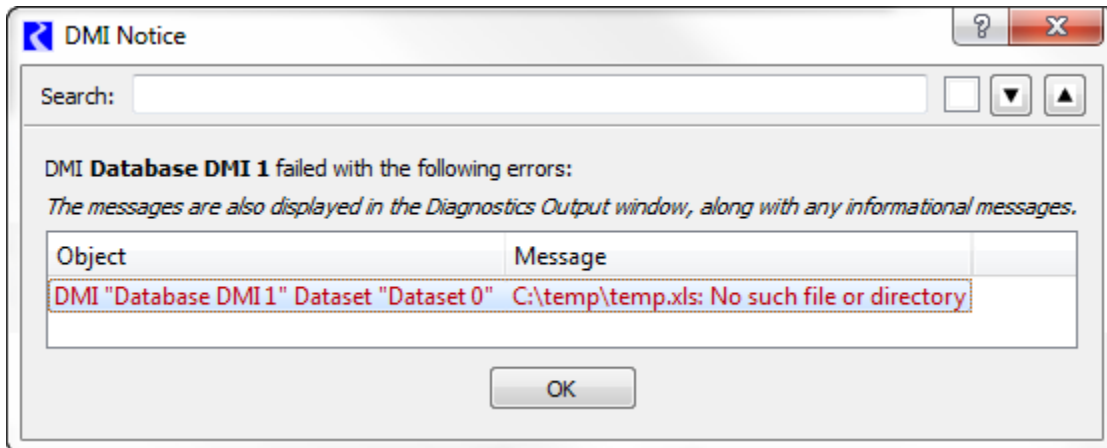


If the DMI is configured to **Confirm Warnings** (discussed [HERE \(Section 4.2.1\)](#) and [HERE \(Section 3.5.3\)](#)) or an error occurs, a separate dialog called the **DMI Notice** will open. This dialog shows a searchable window containing the warning or error messages. Searching is available using the same approach used by diagnostics as described [HERE \(Diagnostics.pdf, Section 1.4.4\)](#).



If the messages are just warnings and you wish to continue execution of the DMI (**Confirm Warnings** not selected), select **OK**. Otherwise, select cancel to stop execution.

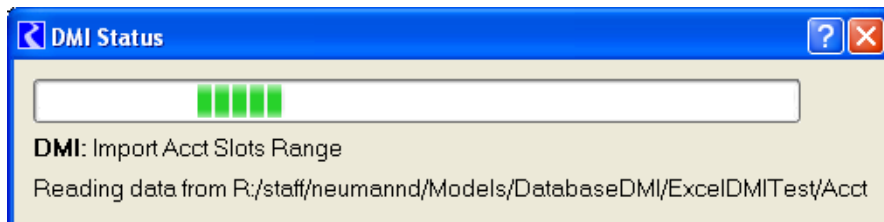
Any errors generated by the DMI are displayed in the DMI Notice window and in the Diagnostic Output dialog in red text as shown in following screenshot. Note, after an error, the DMI stops executing.



Note, there are diagnostics available for DMIs. More information can be found [HERE \(Diagnostics.pdf, Section 5\)](#).













## 2.5.2 DMI Status

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. 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. It will close when the DMI is finished. Because this dialog has minor impact on performance, especially when processing a large number of slots, it is optional and by default is not shown. This status dialog can be shown by toggling the menu **DMI** ➔ **Show Status Dialog** from the **DMI Manager**. This toggle setting is saved with the model file. A sample is shown in the screenshot.













## 2.6 Icon Quick Reference


Icon	Meaning
	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
	The DMI Manager dialog is unlocked
	The DMI Manager dialog is locked

#### DMI Parameter Dialog

	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

#### Database DMI

	Add a new Name Map
---	--------------------

Icon	Meaning
	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.

---

# Control File-Executable Approach

---

## 3. Control File-Executable Approach

### 3.1 Introduction

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:

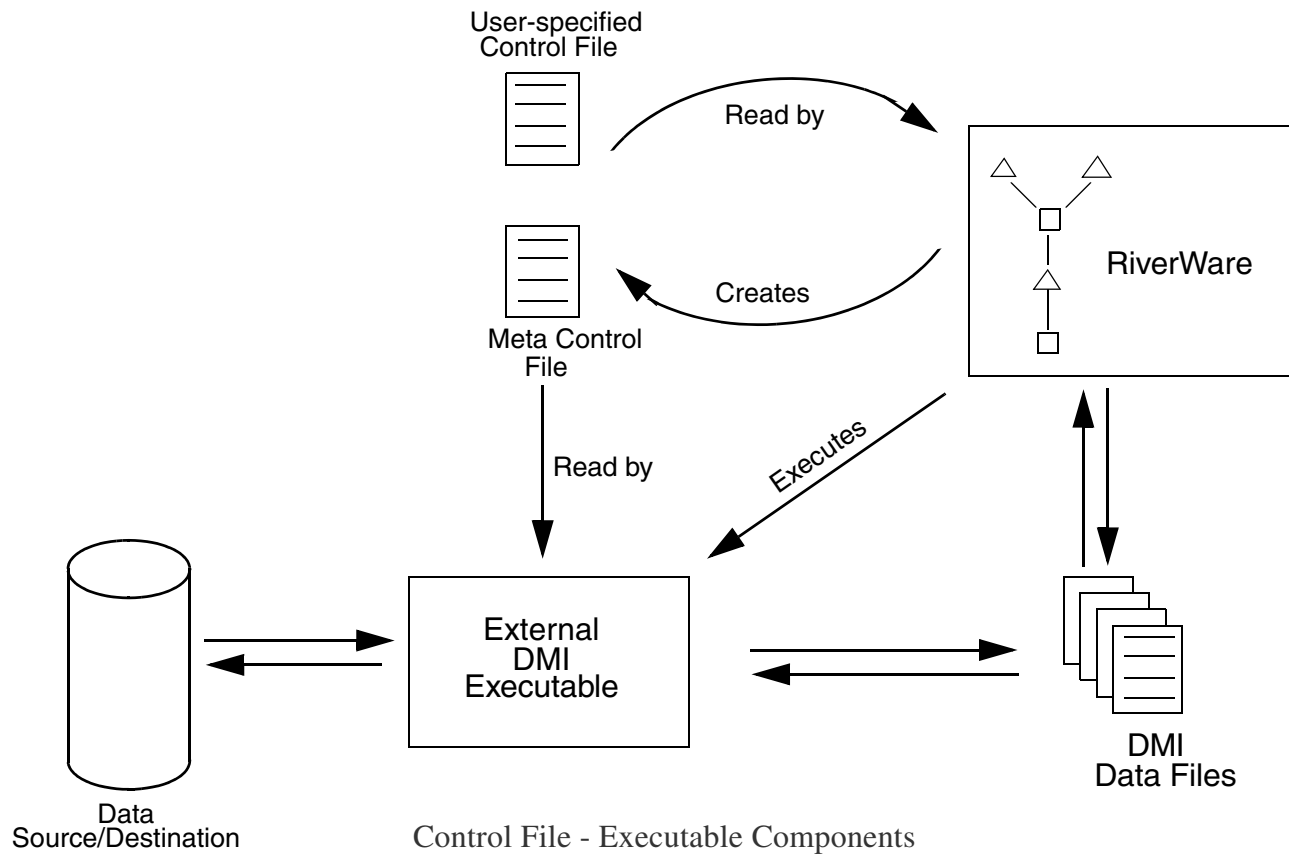
- An import DMI, which transfers data from an external data source to RiverWare.
- An export DMI, which transfers data from RiverWare to an external data sink.

The RiverWare DMI consists of several components. Please refer to the following figure 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 DMI's, this external executable provides the data files which RiverWare loads into the model. For output DMI's, 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 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.

In this figure, 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.
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 writes the Meta Control File which has a fully specified line for each slot in the Control File.
4. RiverWare then runs the External DMI Executable.



5. The executable reads the Meta Control File and interacts with the Data Source (i.e. the database) and creates the DMI Data Files. Alternatively, the user can create the DMI Data Files separately without using an executable. In this case, the executable does not need to do anything.
6. Once the executable is complete, RiverWare reads the DMI Data Files and places data in the slots.

### 3.2 Control File

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

- 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 ...
```

### 3.2.1 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 wildcard 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. More information can be found [HERE \(Accounting.pdf, Section 6.1\)](#).

The above slot specifications lead to twenty six ways of specifying the object.slot pairs:

```
<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

```

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.

### 3.2.2 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 which 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 something like:

```
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 RiverWare units file.

- 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. Since the setting in the user interface 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).

- import=<fixed>|<resize>

The type of import. <fixed> leaves the size of the data series unchanged; <resize> resizes the data series to fit the data. The default is <fixed>

- flags=<true>|<false>

The flags keyword specifies whether or not SeriesSlot 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...

- precision=<display>|<model>

The precision of the data the slot will import or export. The <display> precision is that specified in the slot’s Configuration dialog. The <model> precision is 12, the same precision as all internal calculations. The default is <display>.

- 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.

- slot\_anno = <true>|<false>

The slot\_anno keyword indicates whether to import or export series slot notes ([HERE \(Slots.pdf, Section 5.7\)](#)). The default value for slot\_anno is false, meaning no Notes will be written.

**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

```
#
# 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
```

### 3.3 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 being imported or exported.

```
units: units
units: acre-feet/month
```

- The scale of the data being imported or exported.

```
scale: scale
scale: 100
```

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

```
set_units: new_units
set_units: cms
```

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

```
set_scale: new_scale
set_scale: 10
```

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

```
set_min: new_min
set_min: 10
```

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

```
set_max: new_max
set_max: 1000
```

- 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\_inter: <new\_method>  
<new\_method> is one of lookup, interpolate.  
periodic\_data\_inter: 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). As with the control file, the data files may contain comments (lines beginning with #).

Series slot notes ([HERE \(DMI.pdf, Section 3.5\)](#)) 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.

### 3.4 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. The DMI executable is passed the following command line arguments:

#### 3.4.1 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). It should use the file specification to determine to where to write the data (import DMI), or from where to read the data (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.

### 3.5 DMI Editor Dialog

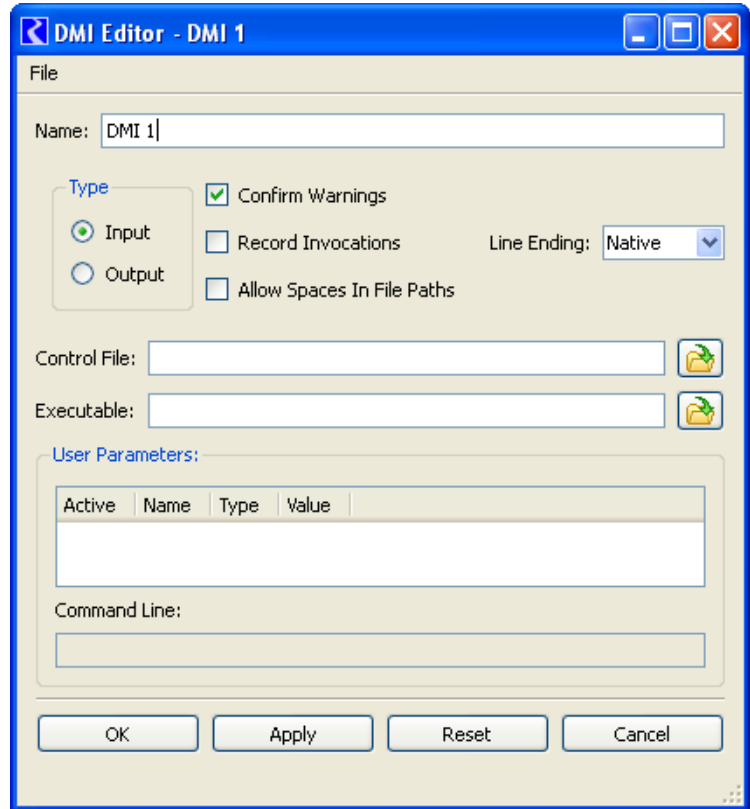
The **DMI Editor** dialog (shown) enables the user to edit the DMI's name, type, confirm warnings, record invocations, allow spaces in file paths, line ending, control file, executable and user parameters.

#### 3.5.1 Name

To edit the DMI's name the user clicks in the **Name** field to edit the name in place:

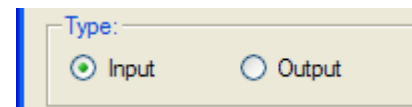
When the edits are applied the DMI's new name is validated; the possible errors are:

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



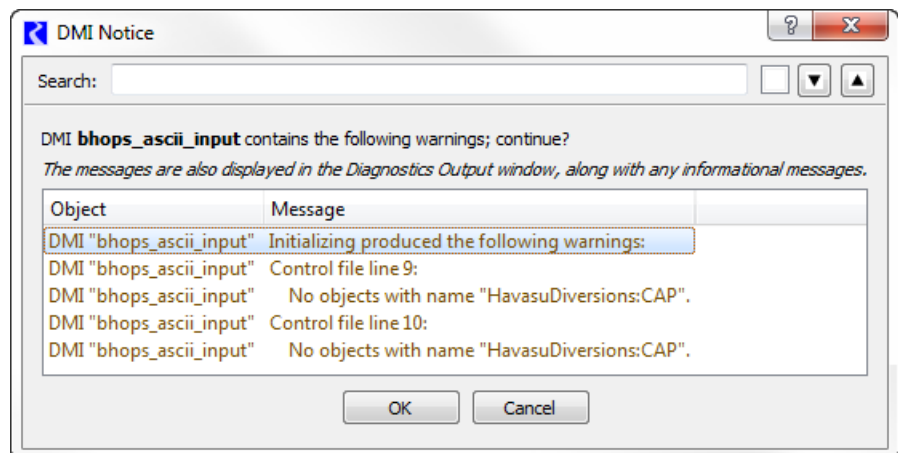
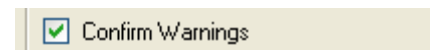
#### 3.5.2 Type

To edit the DMI's type the user selects the appropriate **Type**:



#### 3.5.3 Confirm Warnings Toggle

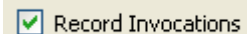
The **Confirm Warnings** toggle controls whether you wish for the DMI to continue when warnings are generated. If **Confirm Warnings** is checked, the user is prompted to confirm any warning messages that appear; if not checked, the DMI continues after any warnings. Following is a sample dialog that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



### 3.5.4 Record Invocations

The **Record Invocations** check box allows RiverWare to maintain information which chronicles input DMI invocations. With this information, values set by DMIs are clearable 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's invocations and this box should remain unchecked. (The information occupies memory and could, over time, degrade performance.)


 Record Invocations

The checkbox is only enabled for input DMIs and the default is unchecked. For more information on clearing values set by an input DMI click [HERE \(Section 2.4\)](#).

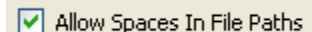
---

**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.

---

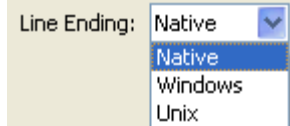
### 3.5.5 Allow Spaces in File Paths

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 that 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.


 Allow Spaces In File Paths

### 3.5.6 Line Ending

The **Line Ending** pulldown 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.


 Line Ending: Native (dropdown menu showing Native, Windows, Unix)

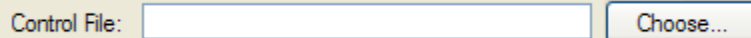
The choices are:

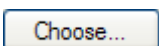
- Native (default, uses the format for the current platform)
- Windows (CR NL)
- Unix (NL)

CR = carriage return, NL = newline

### 3.5.7 Control File

To edit the DMI's control file, the user either:


 Control File:  Choose...

- Selects  to open a file chooser dialog and select the control file

- Clicks in 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

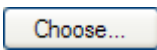
Note that all other validation occurs when the DMI is invoked, including:

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

### 3.5.8 Executable

To edit the DMI’s executable file, the user either:

The screenshot shows a dialog box with a label 'Executable:' followed by a text input field and a 'Choose...' button. The text input field is empty.

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

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

Note that 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

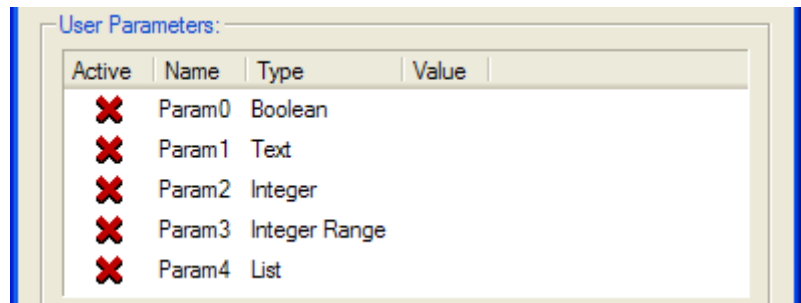
### 3.5.9 User Parameters

To edit the DMI’s user parameters the user specifies:

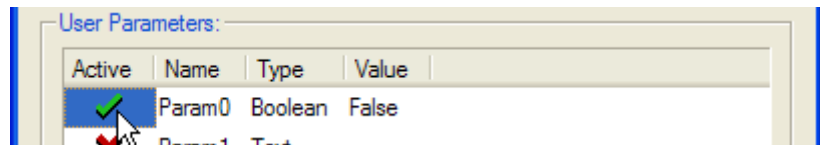
- If a parameter is active (e.g. if it should be included on the DMI executable’s command line)
- If a parameter is active, its value

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

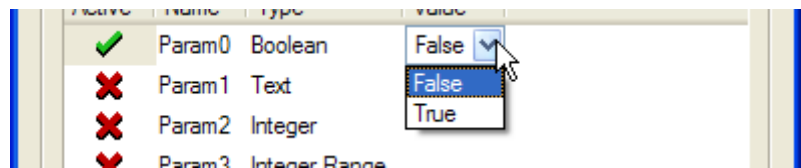
The user toggles a parameter's active state by clicking in the **Active** cell; when a parameter is active its value is displayed in the **Value** cell:



The user edit's a parameter's value by clicking twice in the **Value** cell, activating a control appropriate for the parameter's type:

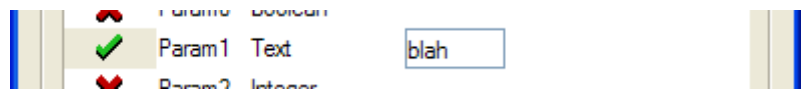


For a **Boolean** parameter the control is a menu containing "False" and "True".

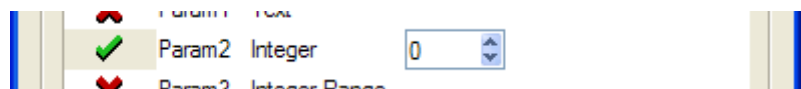


For a **Text** parameter the control is a one line text editor. When the edits are applied the parameter's value is validated; the possible errors are:

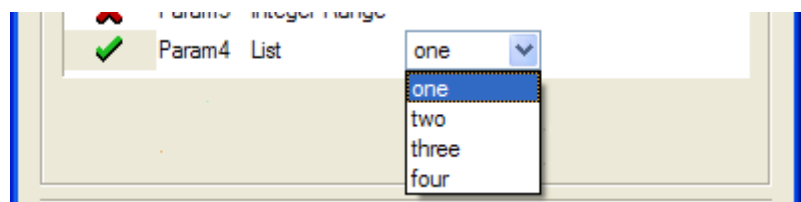
- The value is blank
- The value contains white space



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.

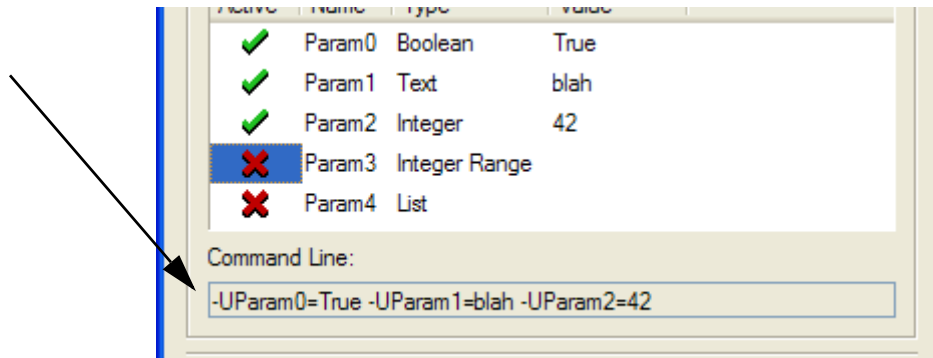


For a **List** parameter the control is a menu containing the list items defined in the DMI Parameter dialog.



### 3.5.10 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).

# Database DMI

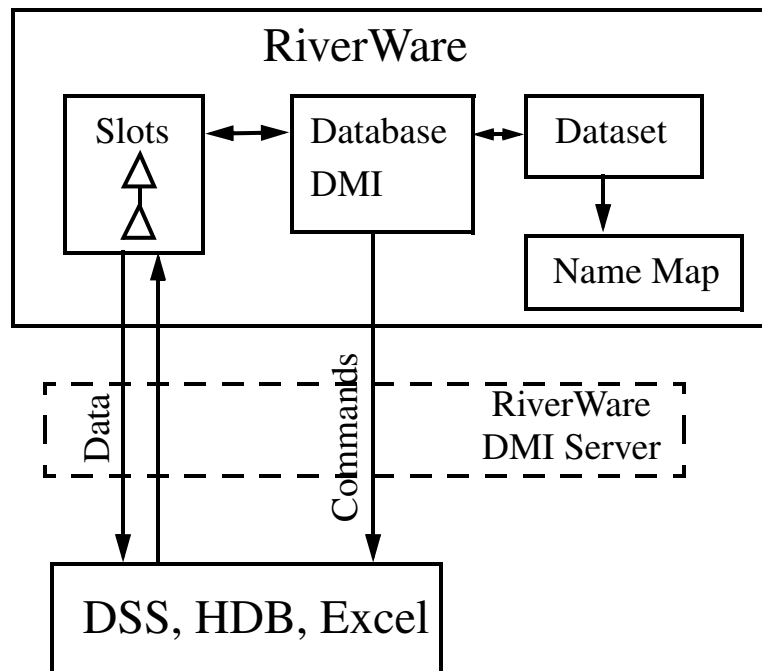
## 4. Database DMI

The second type of DMI, the Database DMI, provides a direct connection between RiverWare and an external database. Databases currently implemented are the U.S. Army Corp of Engineers Hydrologic Engineering Center's Data Storage System (DSS), the Hydrologic Database (HDB), and Excel. DSS supports series data (Series Slots and Table Series Slot columns) and paired data (Table Slots with two columns), while HDB and Excel support series data. This section presents an overview of the Database DMI, then presents details on creating database DMIs, datasets, name maps, and sharing database DMIs.

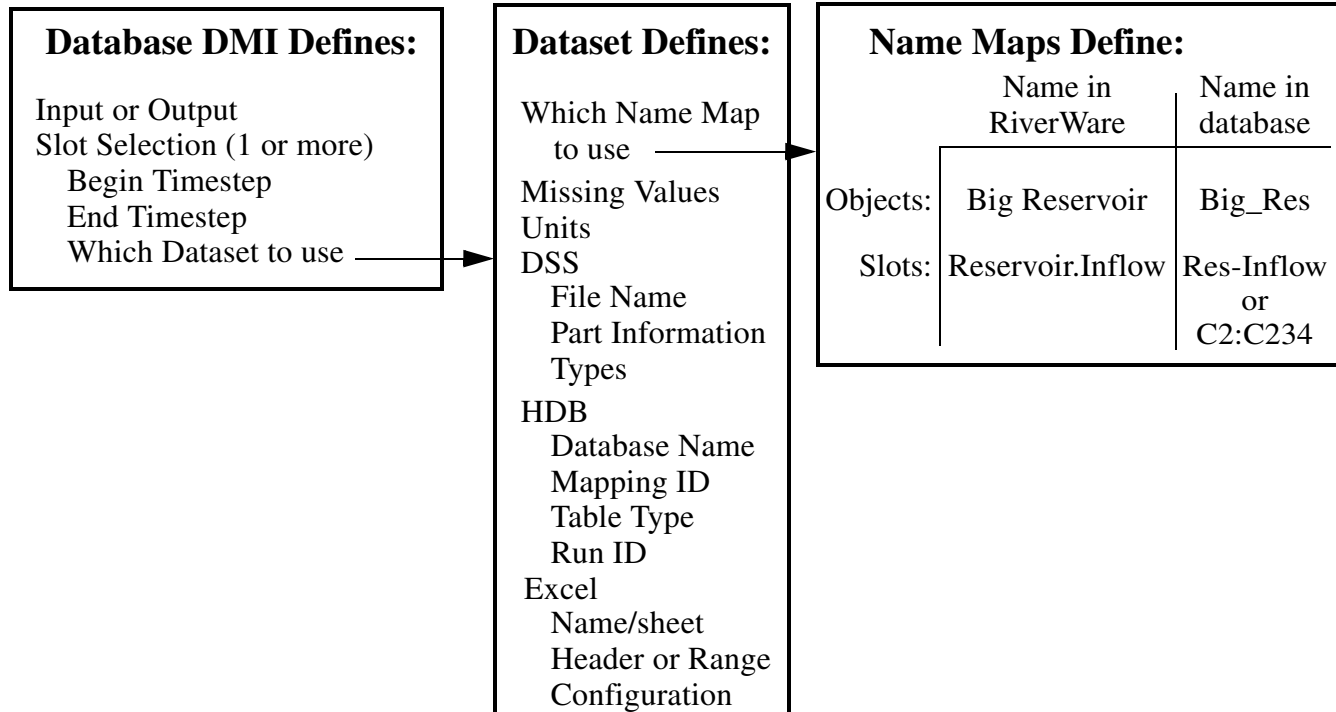
### 4.1 Overview

The Database DMI allows RiverWare to import/export data directly from/to the database. In this document, a "database" refers to a specific database implementation, for example a DSS file. The figure to the right 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.

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.

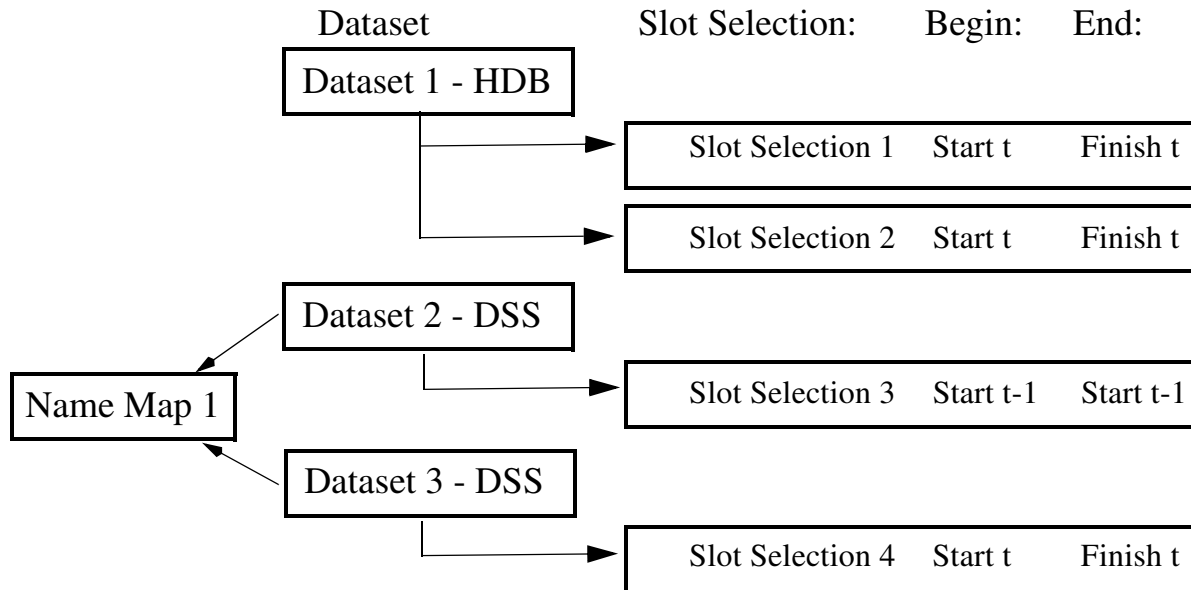


The following figure shows in more detail the three pieces of a database DMI and how they work together to define a DMI. Note, with the DMI Database Editor ([HERE \(page 41\)](#)) you can configure an entire Database DMI in one dialog but these three components still exist and interact with one another.



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 the following figure, 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.



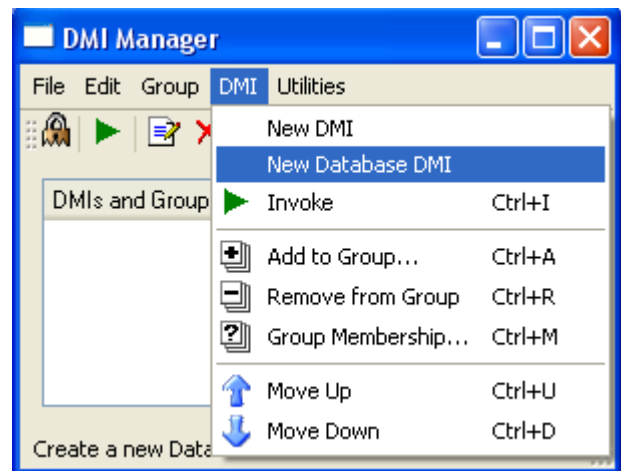
We will now look at each component starting with creating a Database DMI, then look at Datasets followed by Name Maps.

## 4.2 Creating a Database DMI

A Database DMI is created from the main DMI Manager dialog using the **DMI** → **New Database DMI** 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 pulled 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

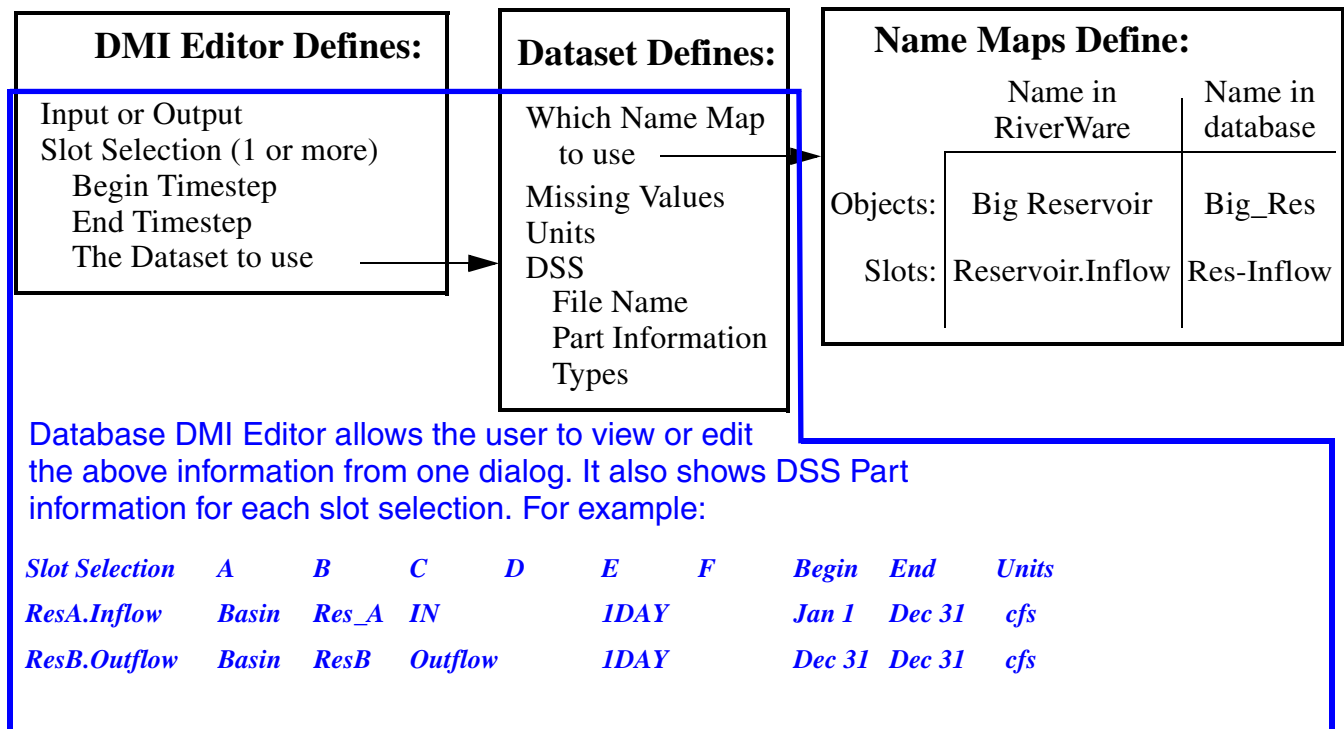
that end, it allows the user to specify all required configuration information and as much optional configuration information as is feasible. The real power of this dialog is that the user can see the fully specified information that will be sent to the database, which is especially useful for DSS.

To open the **Database DMI** dialog, (aka the **New Database DMI Editor**) as shown in the following figure, the user clicks on the **Edit** ➔ **Edit with New Database DMI Editor**.

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 DMI's, 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.

The following figure shows a similar figure to the one presented [HERE \(page 39\)](#) but with the Database DMI Editor also shown. Note, that 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.

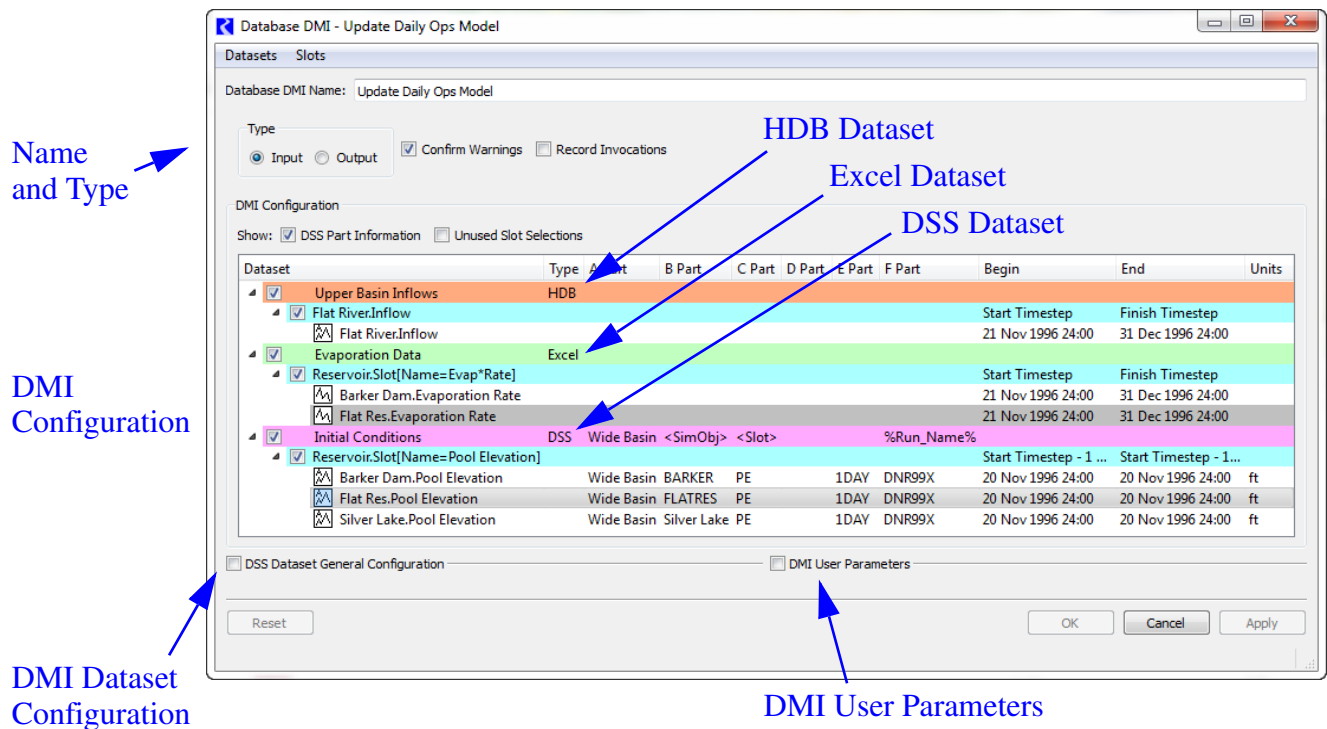


Conceptually the dialog has four sections from the top to the bottom as shown in the following screenshot and described in the following sections:

- Name and Type
- DMI Configuration
- DSS Dataset Configuration
- DMI User Parameters (for DSS)

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

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



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

1. Specify the **Database DMI Name** in the text field.
2. Specify whether the Database DMI is an **Input** or **Output** DMI.
3. Specify if you wish to confirm warnings before the Database DMI continues using the toggle.

4. Specify whether the 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 are clearable 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's 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. For more information on clearing values set by an input DMI click [HERE \(page 15\)](#).

#### 4.2.2 DMI Configuration

The DMI configuration section (middle section) displays the configuration of the DMI. 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.
- See or hide Unused Slot Selections as described [HERE \(page 49\)](#).
- In this dialog, **tool tips** are used to show the contents of cells, the **status bar** in the lower left is used to indicate directions or additional information.

The information is presented in a tree-view with each level representing a more detailed specification as follows:

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.

Name	What it is	Tree Level	Items shown
Slot Selection	Slot Selection Name as shown (or modified) in the Selector dialog. It can include 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.

#### 4.2.2.1 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 as described [HERE \(Enabling and Disabling for Debugging, Page 49\)](#)
- 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. The options are:

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)

Workspace Entity	Description
Exchange	The name of the exchange. (Valid when accounting is enabled)
Payback	The name of the payback. (Valid when accounting is enabled)
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.

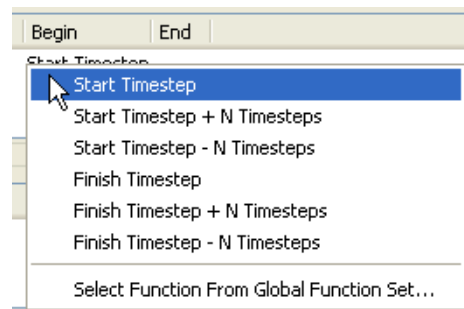
Note, in versions prior to 6.1, Part B was by default the object name and Part C was by default the slot name. Now these need to be explicitly defined for new datasets, existing datasets will be given the <parent> and <slot> entities for parts b and c, respectively.

**User Parameters:** As described [HERE \(DMI User Parameters, Page 50\)](#), you can use parameters in the DSS parts. To use the parameters in the DSS part information, specify the value with the percent sign: %Parameter%

#### 4.2.2.2 Slot Selection

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

- Select slots by clicking on the name of the slot selection. Typically, this is NONE before the slots have been selected.
- Disable a Slot Selection for debugging as described [HERE \(Enabling and Disabling for Debugging, Page 49\)](#)
- 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 left-click 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. So, for example, the user could enter 1:00 April max DayOfMonth, 2006 or 24:00:00



Start Month Start DayOfMonth, Start Year - 24 Hours as a valid datetime (shown in the figure).

Dataset	Type	Begin	End
Evaporation Data	Excel		
Reservoir.Slot[Name=Evaporation Rate]		24:00 Start Month Start DayOfMonth, Start Year - 24 Hours	Finish Timestep
Flat Res.Evaporation Rate		11-20-1996	12-31-1996

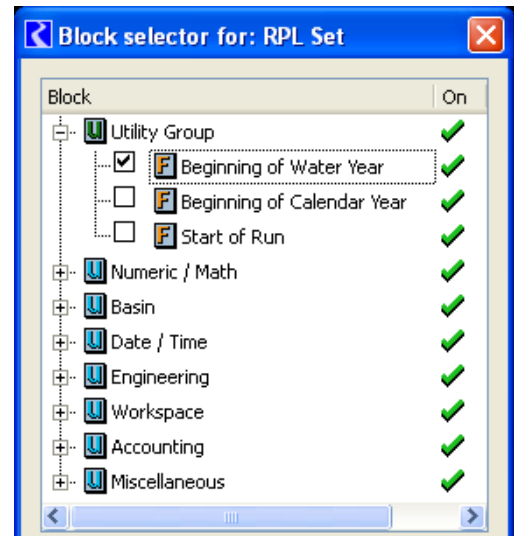
Click [HERE \(RPLTypesPalette.pdf, Section 1.3\)](#) for more information on datetimes. Basically, any fully specified datetime can be used. Note, no @ or quotes "" are necessary when specifying the datetimes in the DMI slots dialog.

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, [HERE \(RPLUserInterface.pdf, Section 1.11\)](#). A block selector opens and displays a tree view of all the global utility groups and predefined function groups. The user then selects a single function by clicking on 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 clicking **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 “( )”.

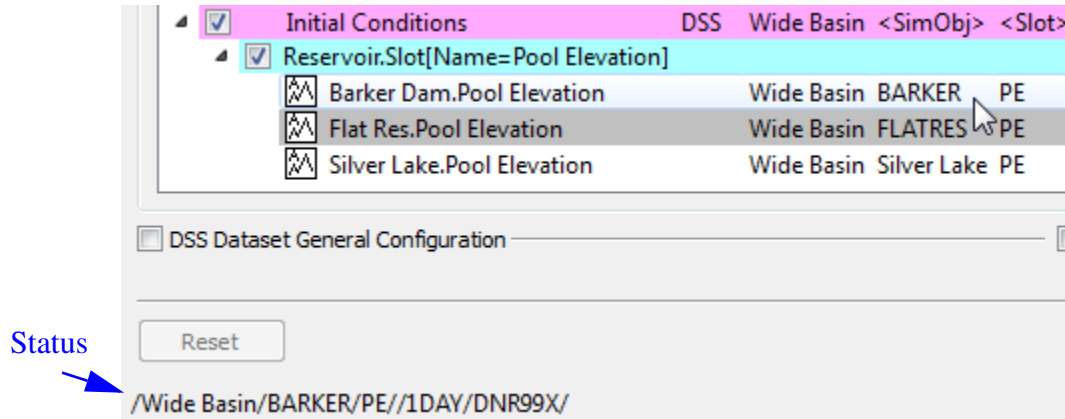


#### 4.2.2.3 Slots in Selection

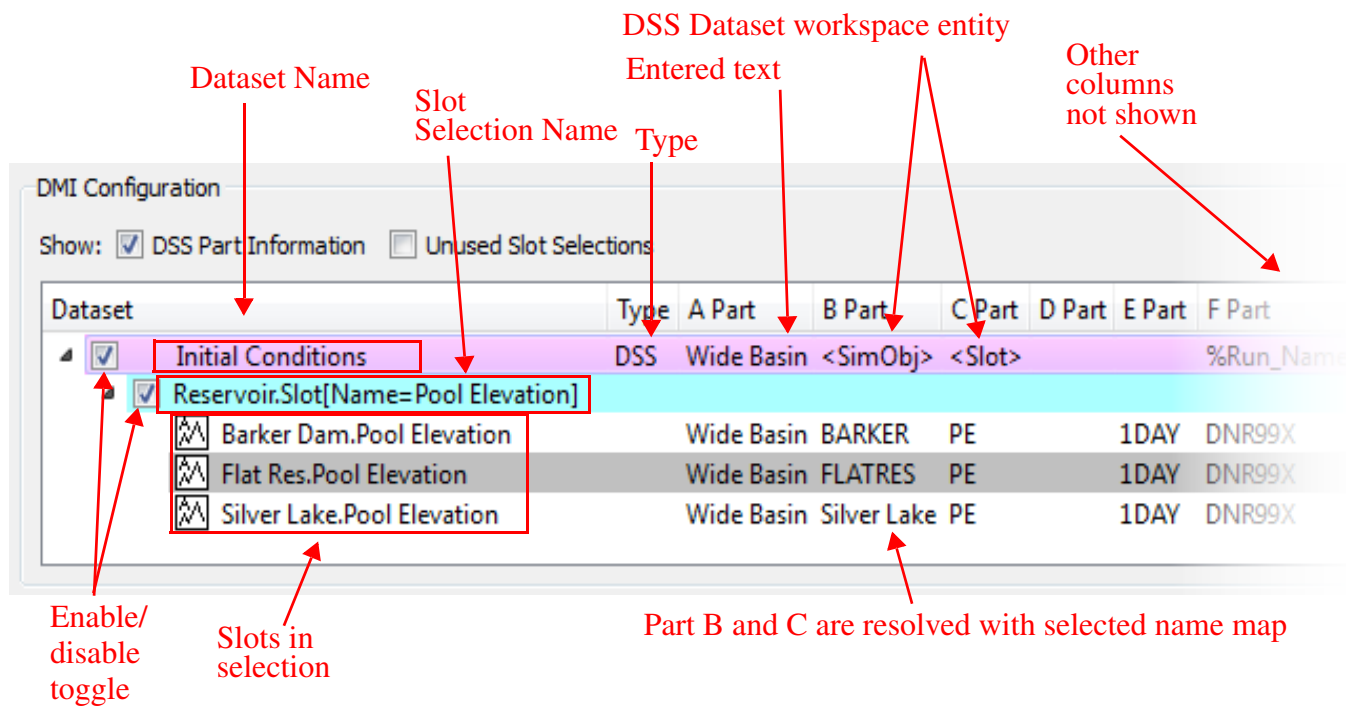
The following actions can be done for Slot Selection rows:

- 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:



Following is a screenshot of a fully defined DSS configuration.



The following screenshot shows the right hand portion of the dialog:

D Part	E Part	F Part	Begin	End	Units
		%Run_Name%	Start Timestep - 1 ...	Start Timestep - 1 ...	
1DAY		DNR99X	20 Nov 1996 24:00	20 Nov 1996 24:00	ft
1DAY		DNR99X	20 Nov 1996 24:00	20 Nov 1996 24:00	ft
1DAY		DNR99X	20 Nov 1996 24:00	20 Nov 1996 24:00	ft

#### 4.2.2.4 Enabling and Disabling for Debugging

Datasets and slot selections can be disabled for debugging purposes. On the left of each Dataset or Slot selection row, there is a toggle. A check mark in the toggle indicates it is enabled. No check mark 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.

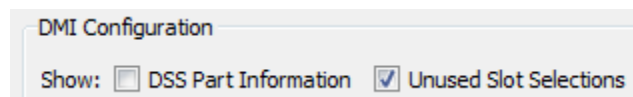
#### 4.2.2.5 Use of Name Maps:

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.

#### 4.2.2.6 Unused Slot Selections

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



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.

Unused Slot Selections			
		Start Timestep	Finish Timestep
<input checked="" type="checkbox"/>	Flat River.Inflow		
	 Flat River.Inflow	11-21-1996	12-31-1996
<input checked="" type="checkbox"/>	Reservoir.Slot[Name=Evap*Rate]		
		Start Timestep	Finish Timestep

### 4.2.3 DSS Dataset General Configuration

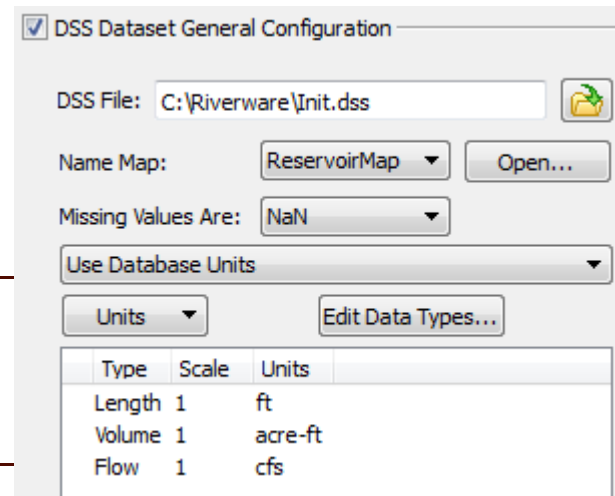
For DSS datasets, an additional panel allows you to fully edit the dataset. Note, this is not available for Excel or HDB. For those datasets, you can double click the dataset and it will open the dataset editor.

For DSS datasets, toggle the **DSS Datasets General Configuration** section (bottom left section) of the dialog. 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 **Open...** the name map manager.
- Specify the missing value behavior for the selected DSS Dataset.
- Specify the units for the selected DSS Dataset.
- Configure Datatypes by clicking the **Edit Data Types...** button.

These fields are described in the DSS Dataset section [HERE \(page 59\)](#).

**Note:** 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.



Type	Scale	Units
Length	1	ft
Volume	1	acre-ft
Flow	1	cfs

### 4.2.4 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**: [HERE \(DMI Parameter Dialog, Page 8\)](#).

Edit a parameter's value by double clicking in 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 on 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. The screenshot to the right shows this example.

The screenshot shows three parts of the Database DMI Editor interface:

- Parameter specification:** A table with columns D Part, E Part, F Part, Begin, End, and Units. The F Part column contains the text "%Run\_Name%".
- Parameter resolved:** The same table below, where the F Part column now contains the resolved value "DNR99X".
- Parameter value:** A table titled "DMI User Parameters" with columns Name, Type, and Value. The "Run\_Name" row has a value of "DNR99X".

## 4.2.5 Examples

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

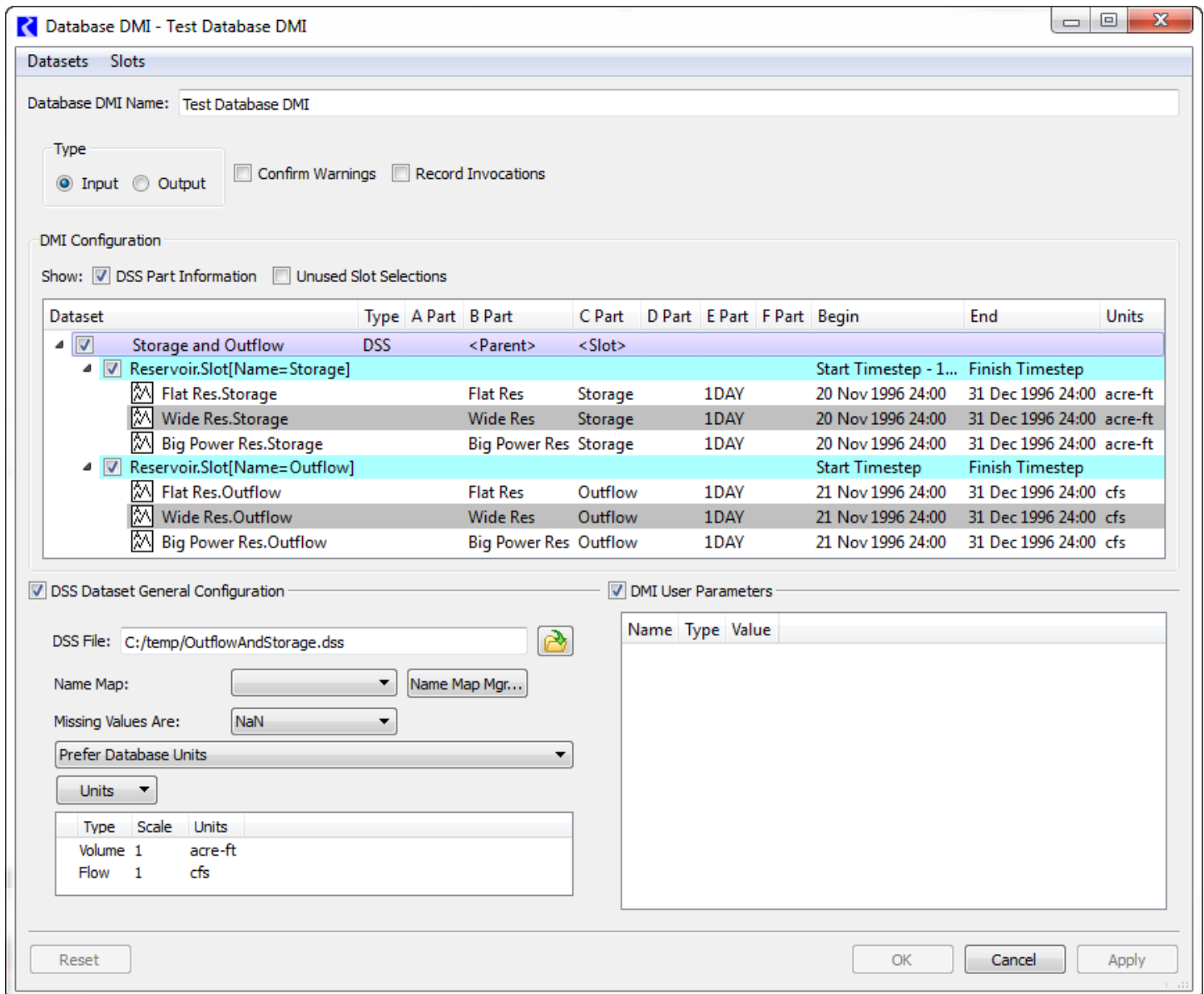
### 4.2.5.1 Example 1: DSS DMI using the Database DMI Editor:

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** ➔ **DMI Manager**.
2. Create a new Database DMI using the **DMI** ➔ **New Database DMI** menu.
3. Open the Database DMI Editor using the **Edit** ➔ **Edit** menu.
4. In the upper portion of the dialog [HERE \(Section 4.2.1\)](#), enter a name, whether it is an Input or Output DMI, whether to confirm warnings, and whether to record invocations. Lets assume we are creating an Output DMI. Leave everything else unchecked.
5. First, add a dataset using the **Datasets** ➔ **New DSS Dataset** menu.
6. Expand the Dataset column and double click on 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 on the dataset you added, then use the **New Slot Selection**. A row is added to the tree view below the dataset.
8. Double click on the word NONE to open the slot selector. Use the selector to choose Reservoir.Storage. Add another slot selection and select Reservoir.Outflow. An example of a similar selection is presented [HERE \(Selector.pdf, Section 7.3\)](#).
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 on 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. Click on 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 clicking on the appropriate column on the Dataset row. For example, we could enter OBS for observed in part F by clicking in that field on the dataset row and typing it in.
13. Now configure the dataset. Select the dataset row. Toward the bottom of the dialog, click the **DSS Dataset General 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 [HERE \(Section 4.2.3\)](#). A name map is created outside of this dialog [HERE \(page 81\)](#). 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 [HERE \(DSS Dataset General Configuration, Page 50\)](#), Click on the Units button and choose New. Repeat to add two rows.
17. Click-pause-click on 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. Click OK to close the dialog and apply the changes. Invoke the DMI from the DMI manager using the **DMI ➔ Invoke** menu.

The following screenshot shows the configuration for this example.

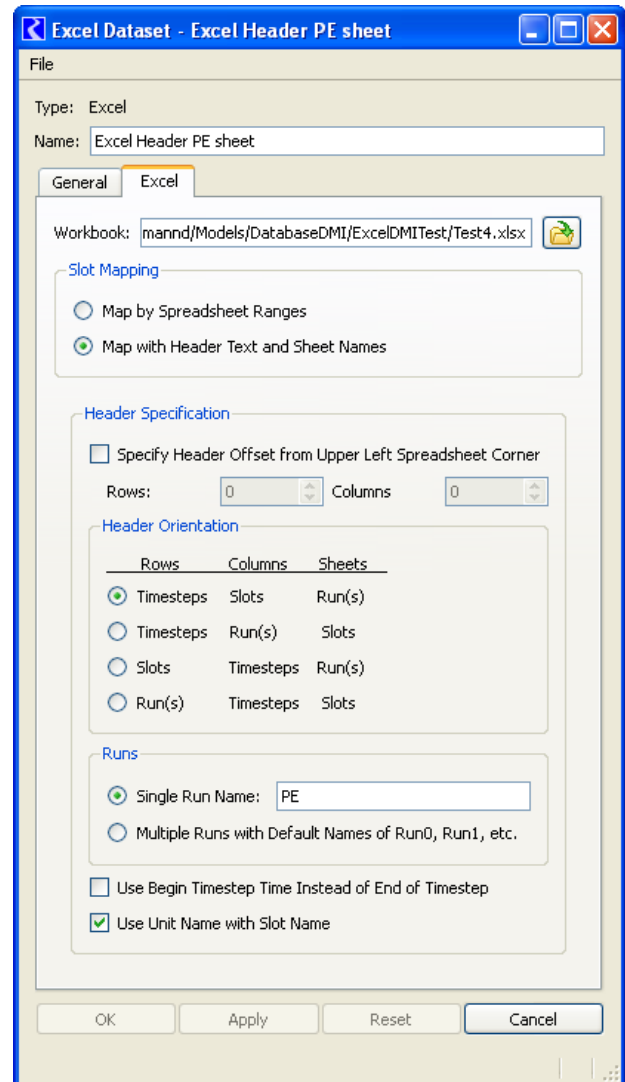


#### 4.2.5.2 Example 2: Excel DMI to export data using Headers

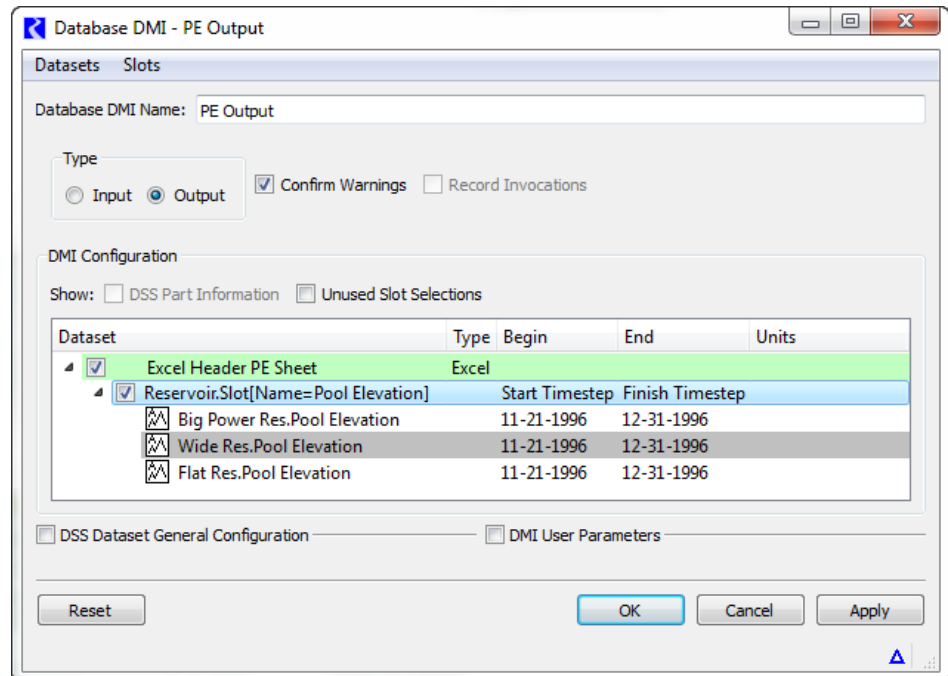
This example shows how to create a Database DMI to export data to Excel using the Header approach. We assume there are no other Datasets or Name Maps already defined. You would take the following steps:

1. From the workspace, open the DMI Manager using the **DMI** ➔ **DMI Manager**.
2. Choose **Utilities** ➔ **Datasets...**

3. Choose **Dataset** ➔ **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. Click on 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** and they should look similar to the samples shown [HERE \(Map by Spreadsheet Range, Page 74\)](#) based on the configuration. Let’s use the header approach (described [HERE \(Map by Header Text and Sheet Names, Page 77\)](#)); choose the **Map by Header Text and Sheet Names** option.
8. Specify the Header orientation and specify a sheet name. In this sample, use the Timestep, Slots, Runs option.
9. Optionally, specify to **Use Unit Name with the Slot Name**. The setup for the Excel tab is shown in the screenshot. With this setup, you do not have to use Name Maps, but you could. Thus, we have a fully configured dataset.
10. Click **OK** to accept the changes and close the dialog
11. Back in the DMI Manager, create a new Database DMI using the **DMI** ➔ **New Database DMI** menu.
12. Open the Database DMI Editor using the **Edit** ➔ **Edit** menu.
13. In the upper portion of the dialog [HERE \(Name and Type, Page 43\)](#), enter a name, that it will be an Output DMI, and whether to confirm warnings.
14. Now add the dataset you configured earlier. Choose **Datasets** ➔ **Add Existing...** to open the Dataset Selector dialog.
15. Choose the “Excel Header PE Sheet” dataset and click OK.
16. 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** ➔ **New Slot Selection** menu.



17. Double-click on the word NONE to open the slot selector. Use the selector to choose all Reservoir.Pool Elevations. An example of a similar selection is presented [HERE \(Selector.pdf, Section 7.3\)](#).
18. 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”.
19. The following screenshot shows this setup:
20. This database DMI is now fully configured. You can execute it from the DMI manager. Here is the result for our three reservoir model:



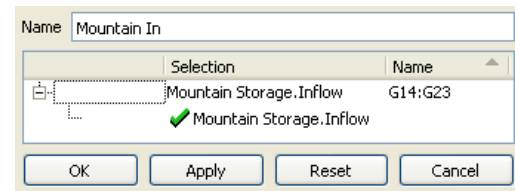
	A	B	C	D
1		Big Power Res.Pool Elevation (ft)	Wide Res.Pool Elevation (ft)	Flat Res.Pool Elevation (ft)
2	11/21/1996	1192.208238	635.1760648	NaN
3	11/22/1996	1192.241859	635.1318182	NaN
4	11/23/1996	1192.302769	634.934238	NaN
5	11/24/1996	1192.335546	634.9504454	NaN
6	11/25/1996	1192.365509	635.0791799	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

### 4.2.5.3 Example 3: Excel DMI to import Data using Ranges

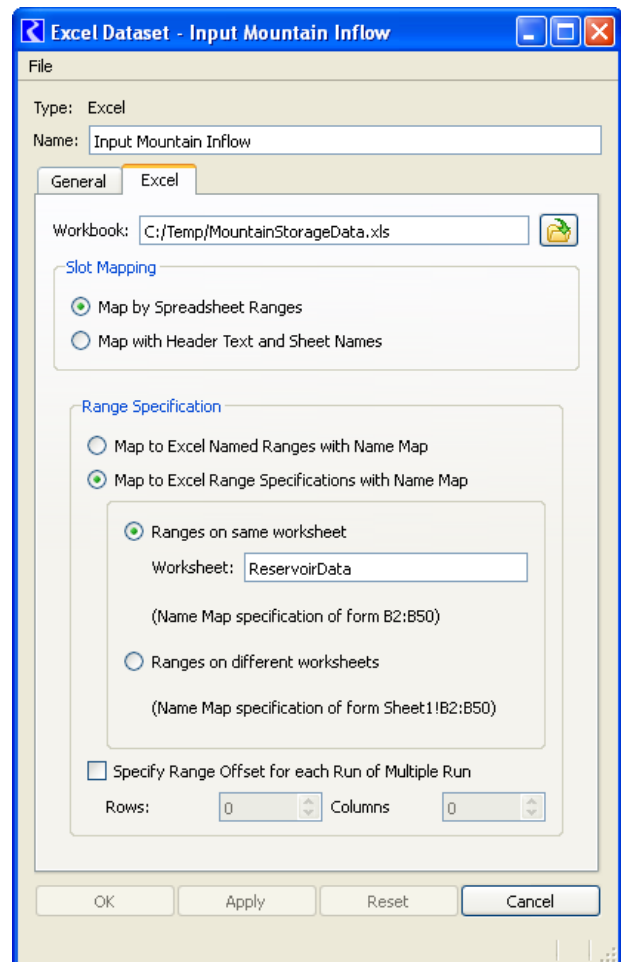
This example shows how to create a Database DMI to import data from Excel using the Ranges approach. We assume there are no other Datasets or Name Maps already defined. You would take the following steps:

- Following 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.
- From the workspace, open the DMI Manager using the **DMI** ➔ **DMI Manager**.
- From the DMI Manager choose **Utilities** ➔ **Name Maps...**
- Choose **Name Map** ➔ **New...** or click the + button.
- Double click the newly created Name Map.
- Change the name to “Mountain In”
- Click **Name Map** ➔ **New...** to add a row to the list.
- Click-pause-click on the word **NONE** and choose **Select Slot**.
- Using the selector, choose the reservoir slot **Mountain Storage.Inflow**. Click **OK**.
- In the Name Map, in the **Name** column type “G14 :G23” (without quotes).
- Click **OK** in the Name Map and **OK/Cancel** in the **Name Map Manager** to close.
- In the DMI Manager, choose **Utilities** ➔ **Dataset...**
- Choose **Dataset** ➔ **New Excel Dataset...**
- Double click the newly created dataset to open it.
- You will create an input DMI so specify a name in the **Name:** field: “Input Mountain Inflow”.
- In the **Name Map** section, choose the Name Map you created “Mountain In” from the pull down menu.
- Click **Units** ➔ **New**
- Under the **Type** column, click-pause-click and choose **Flow** from the pull down menu.
- Change the **Scale** to 1000 and the **Units** to cfs.

	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							

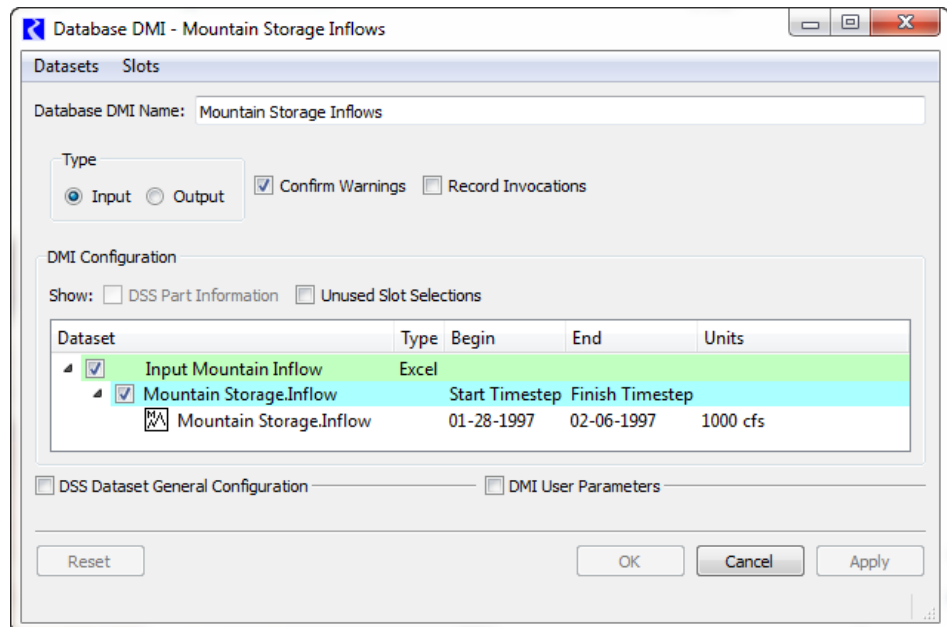
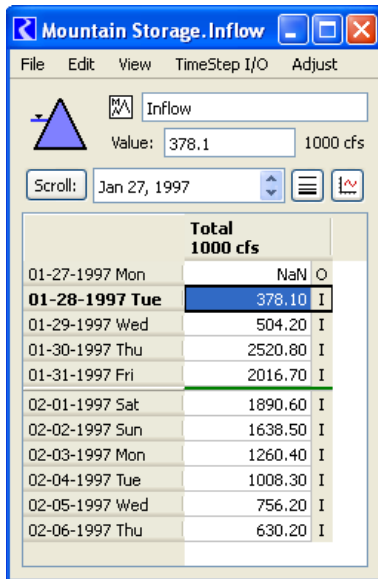


20. Click **Apply**.
21. Click on the **Excel** tab and type in or navigate to the worksheet where the data resides. In the sample, it is in C:/Temp/MountainStorageData.xls.
22. Click the **Map by Spreadsheet Ranges** button.
23. Choose the **Map to Excel Range Specification with Name Map**.
24. The 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. Click **OK** to apply and close the dialog.
26. Back in the DMI Manager, create a new Database DMI using the **DMI ➤ New Database DMI** menu.
27. Open the Database DMI Editor using the **Edit ➤ Edit** menu.
28. In the upper portion of the dialog [HERE \(Name and Type, Page 43\)](#), enter a name and that it will be an **Input DMI**.
29. Now add the dataset you configured earlier. Choose **Datasets ➤ Add Existing...** to open the Dataset Selector dialog.
30. Choose the **Input Mountain Inflow** dataset and click OK.
31. We will select the slots that we wish to input. With the Input Mountain Inflow row highlighted, select **Datasets ➤ 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.

35. This database DMI is now fully configured. You can execute it from the DMI manager. The result is shown in the following slot:

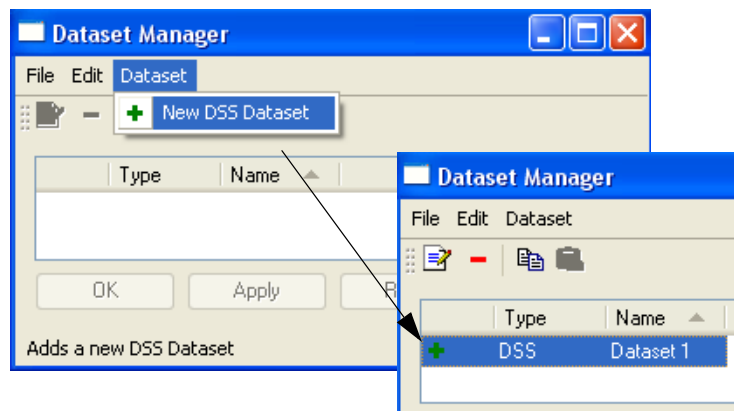


# Datasets

## 4.3 Datasets

A dataset refers to a subset of a database and is specific to the type of database. It 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 the following screenshot. The Dataset Manager is accessed from the **Utilities** ➔ **Dataset...** menu on the **DMI Manager**.



Selecting **Dataset** ➔ **New DSS Dataset**, for example, adds a new DSS dataset. This adds the new dataset to the manager as shown in screenshot below, right.

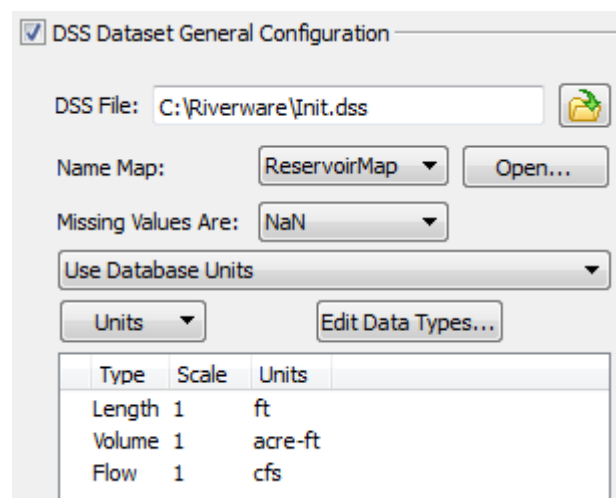
### 4.3.1 DSS Datasets

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 **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 ([HERE \( page 50\)](#)) 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.

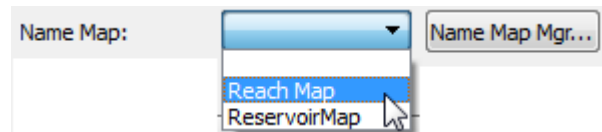
The DSS file name text shows the user parameters while a tooltip shows the substituted values. The



user parameters can be all or part of the DSS file name and can be used in conjunction with environment variables, for example:

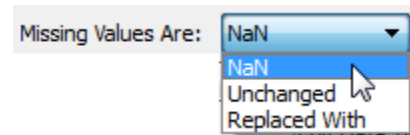
- A single parameter: %DssFile%
- A path and a parameter: C:\model\data\%DssFile%
- Multiple parameters: %ModelDir%\data\%DssFile% or %DataDir%\%DssFile%.dss
- Environment variable and a parameter: \$(DATA\_DIR)\%DssFile%.dss

**Name Map:** Optionally, select the desired Name Map from the **Name Map:** drop down menu as shown in the following screenshot. The selections are based on the Name Maps configured in the **Name Map Manager**. For more information on Name Maps click [HERE \(page 81\)](#). If you have no name map selected, Click 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

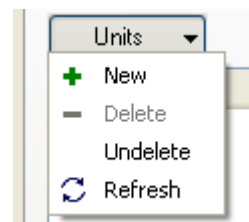
- **NaN:** RiverWare default, missing values are NaN in the database
- **Unchanged:** The DMI will not change missing values on import, uses NaN's on export
- **Replace With:** Provides a user input value that is substituted on import and export



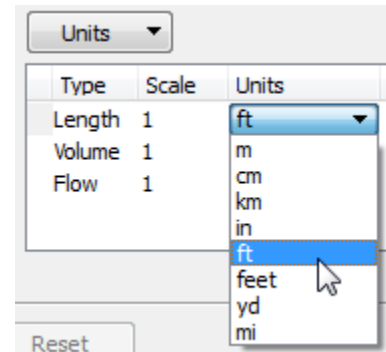
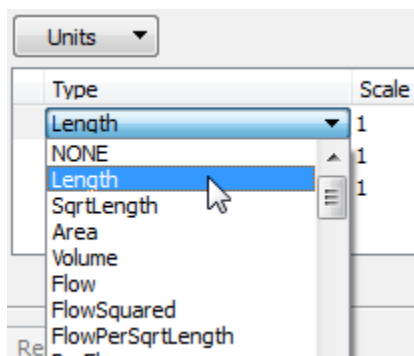
**Units:** The units that are used in the DSS database are specified in the **Dataset** dialog. A pull-down menu allows the user to specify that the Database DMI should **Use Database Units**, **Prefer Database Units**, or **Use Dataset Units**. A screenshot is shown in the following figure.

- **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.

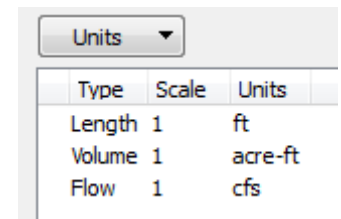
If the user chooses either of the second two options, the user must specify the units in the window below the pull-down menu. Click on the **Units** button then the **+** icon to add a new unit type:



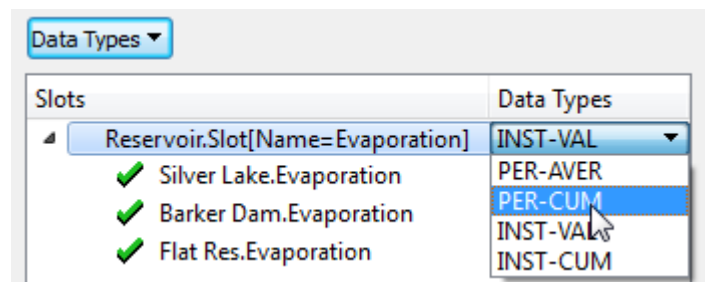
This adds an entry to the list. Click on the word **NONE** under the **Type** to change the type. A screen shot of this is shown in the following figure. Change the **Scale** and **Units** in a similar manner. These units are used to specify the units that will be written to the database if an Output DMI is used and the units that the database contains if an Input DMI is used.



An example of specified units is shown:



**Data Types:** Click the Edit Data Types button to edit the data types. This opens a dialog as shown in the following screenshot. 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

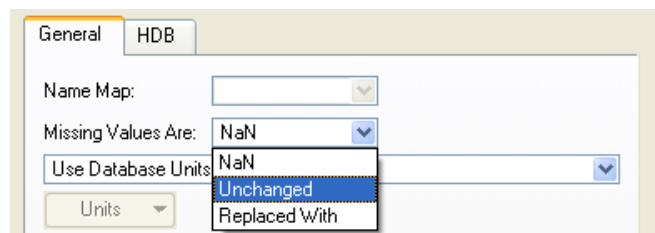
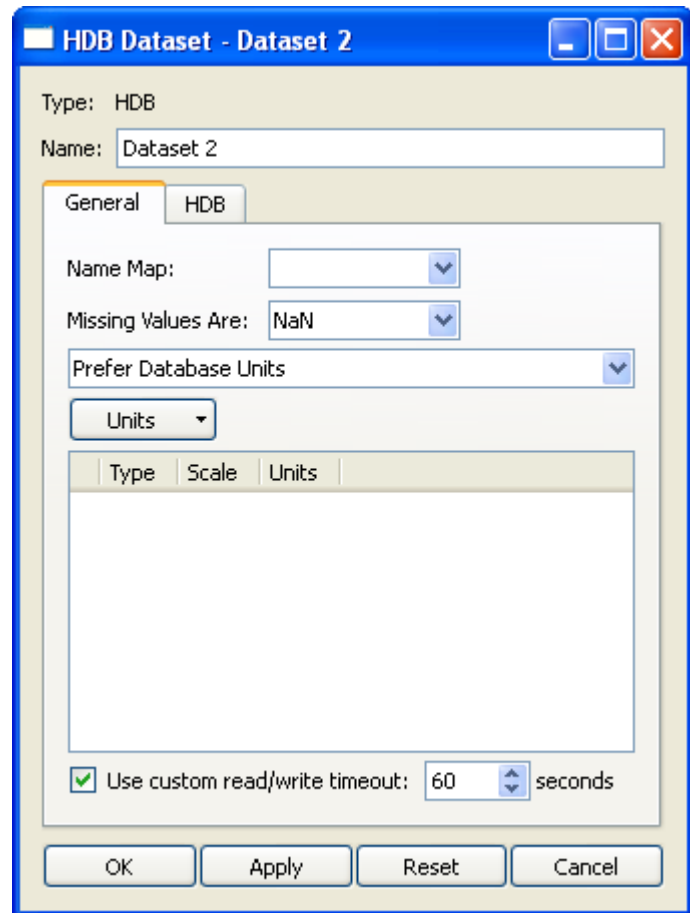
## 4.3.2 HDB Datasets

Double clicking on 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. Following is a description of the configuration options.

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.

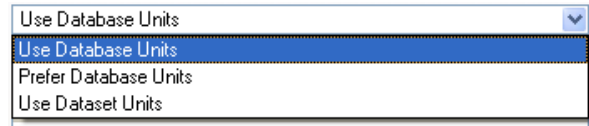
The user specifies how missing values are handled when exchanging data with HDB. In general, a NaN in RiverWare is not represented as a data record in HDB. The current choices are

- **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.



A pull-down menu in the **Dataset** dialog allows the user to specify how units should be handled for a dataset. The choices are **Use Database Units**, **Prefer Database Units**, or **Use Dataset Units**. Units are defined in HDB for each piece of data, so **Use**

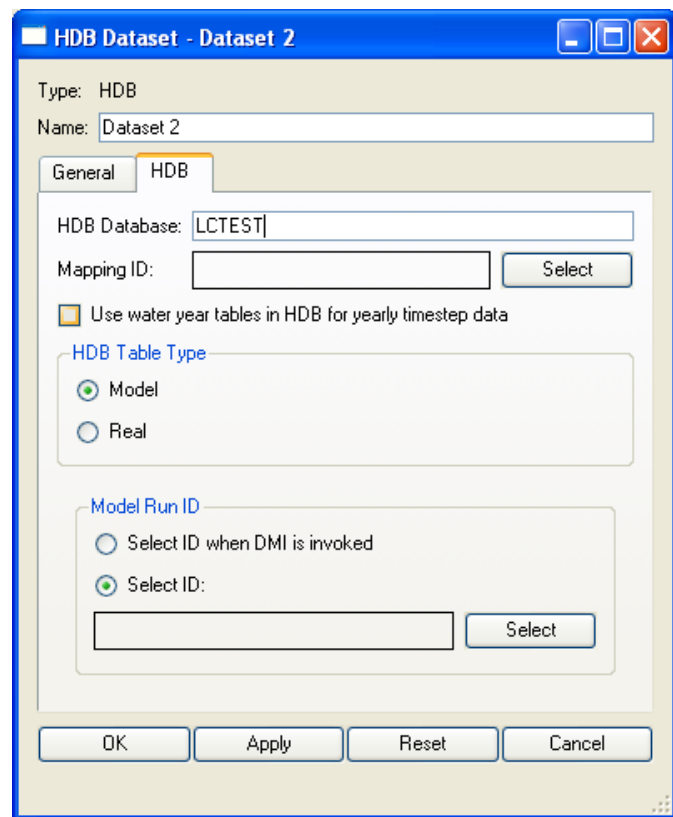
**Database Units** is always applicable. If the unit type of the data in HDB does not match the unit type of the slot in RiverWare and the units cannot be converted, the slot is skipped and a warning message is given. The exception is if the unit type of the slot in RiverWare is NOUNITS. Here the unit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



Also, in the general tab, the user can specify to **Use custom read/write timeout** and specify a time value. The default is 60 seconds. This value is used to specify the amount of time the DMI should wait until it considers the connection between RiverWare and the database dead. Sometimes, the default of 60 seconds is not enough if there is a lot of data being transferred.

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

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. The user needs to have Oracle networking available on their computer through Oracle Client or a full Oracle database installation.

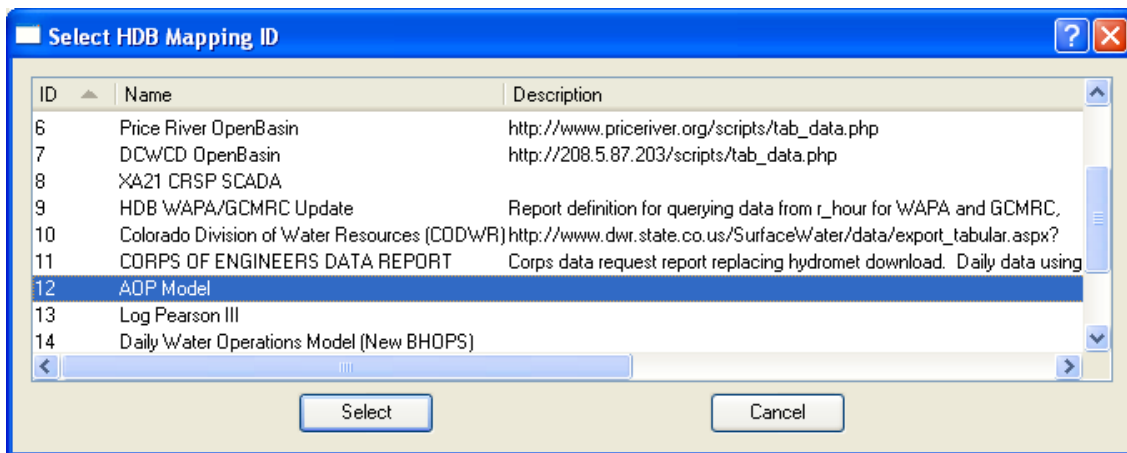
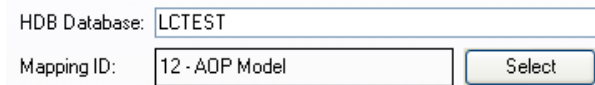
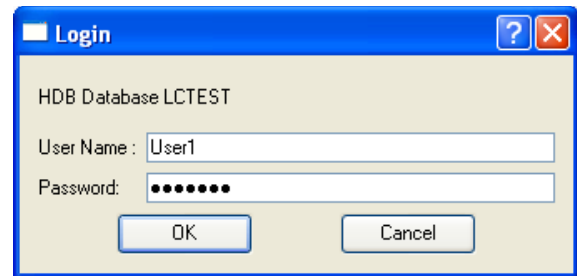


The first time in a RiverWare session that a specified database is connected to, the user is prompted to provide a user name and password to log in. For a user to retrieve metadata or read and write data to HDB, they must be granted the roles `model_priv_role` and `app_role` in the database to have the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

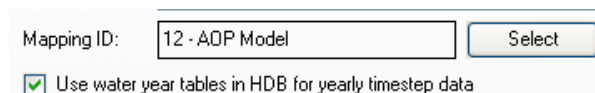
To run in batch mode, the login information must be provided in a file. The code looks for a file named `.dblogins` in the user's home directory, which is specified by the `HOME` environment variable on Solaris, and the combination of the automatically defined `HOMEDRIVE` and `HOMEPATH` environment variables 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 the user wants 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.

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. The user must click 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. The user 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 yearly 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 yearly timestep data** has been provided. If a user checks this box, all yearly 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.

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

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 the user in the **Real** frame that appears with the selection of a **Real** table type in the dataset dialog.

Required for Writing to Real Tables

Agency ID:

Collection Sys ID:

Use Overwrite Flag

HDB Dataset - Dataset 2

Type: HDB

Name:

General | HDB

HDB Database:

Mapping ID:

Use water year tables in HDB for yearly timestep data

HDB Table Type

Model

Real

Required for Writing to Real Tables

Agency ID:

Collection Sys ID:

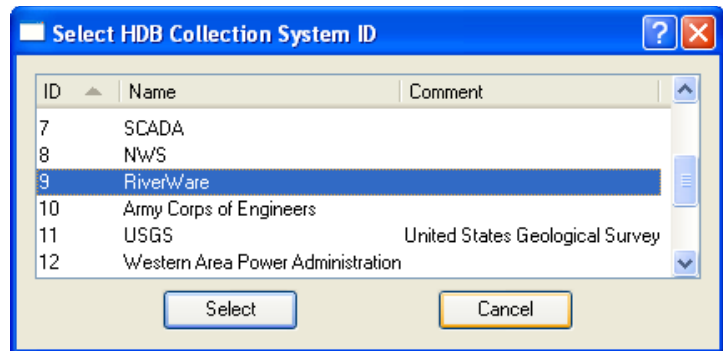
Use Overwrite Flag

The **Agency ID** field specifies what agency ID will be associated with the data when it is written to the real tables. The user must select an agency ID by clicking 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. The user can then select an agency, which populates the **Agency Id** field on the dataset dialog with the ID and name of the agency.

Select HDB Agency ID

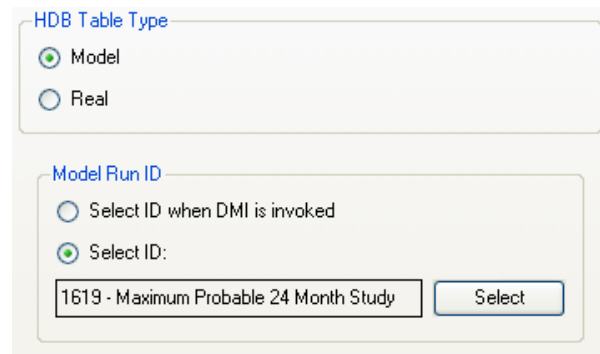
ID	Name	Abbreviation
5	Uintah Water Conservancy District	
6	Collbran Conservancy District	
7	Bureau of Reclamation	USBR
8	Tri-County Water Conservancy District	
9	Dolores Water Conservancy District	
10	Eden Valley Irrigation And Drainage District	

The **Collection Sys ID** field specifies what collection system ID will be associated with the data when it is written to the real tables. The user must select a collection system ID by clicking 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. The user 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.



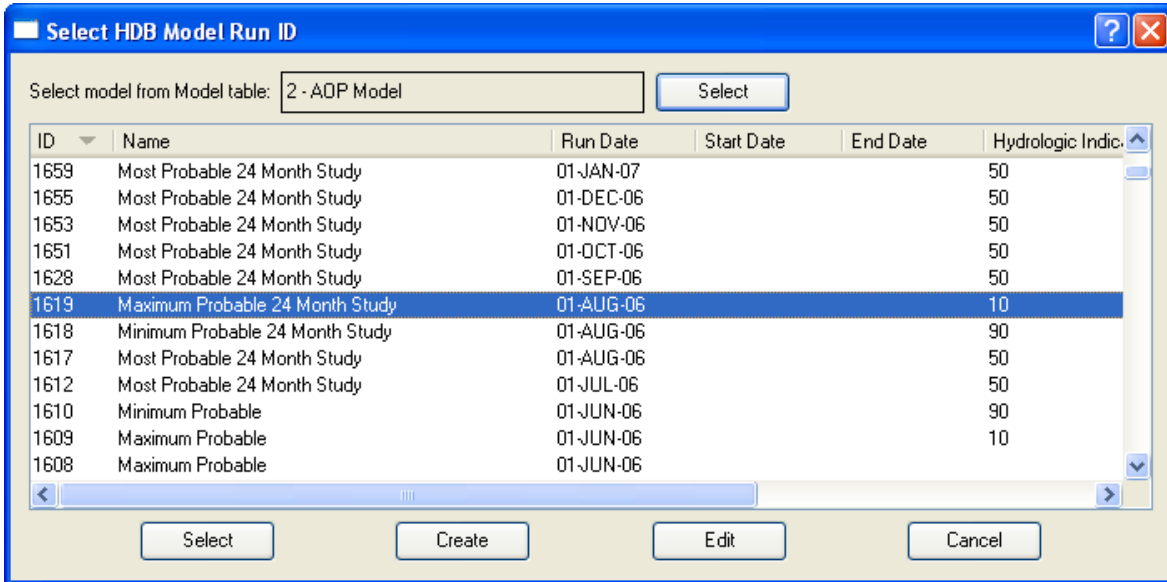
The **Use Overwrite Flag** check box specifies whether or not data is written to the real tables with an overwrite flag. This flag has some special meaning for how the data is handled in the database, namely that when the value is written to its real table, it cannot be overwritten by aggregations of data from a shorter interval. Normally this box would not be checked.

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.

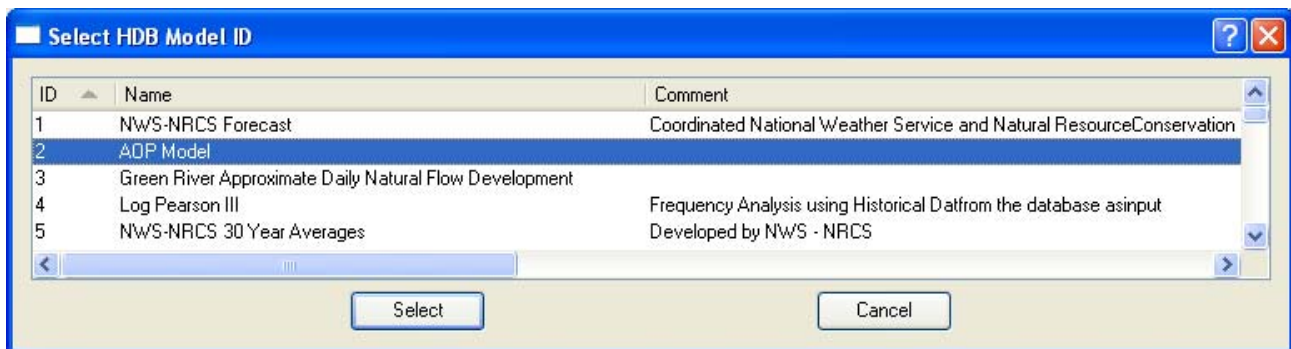


The **Model Run ID** frame has radio buttons for the user to **Select ID** or **Select ID when DMI is invoked**. The **Select ID** option allows the user to preselect a model run ID to associate with the dataset for

reading and writing data with the model tables. To select a `model_run_id` the user must click 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. Clicking 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.



The user 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. The user 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, the user can also edit an existing ID or create a new one. To edit or create a model run ID the user 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 clicking 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 the user to edit these fields. Clicking 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 the user 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. Clicking **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 clicking any portion of the date and typing or using the spinners. At least the required fields marked with an asterisk must be filled out. Clicking 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, the radio button for the user to **Select ID when DMI is invoked**. When this option is selected, the user does not preselect a model run ID that is always used with the dataset, but rather is 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 the user to accidentally overwrite the previously saved data by forgetting to change the ID, the user 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 the user 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 that if the same dataset is used multiple times in a DMI, the user is 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.

# Excel Datasets

## 4.3.3 Excel Datasets

Excel datasets provide user configuration settings that specify how you wish the Database DMI to work with Excel. Excel datasets can only be used on the Windows platform and Excel must be available on the system. RiverWare actually starts a non-visible copy of Excel in the background and accesses 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.

---

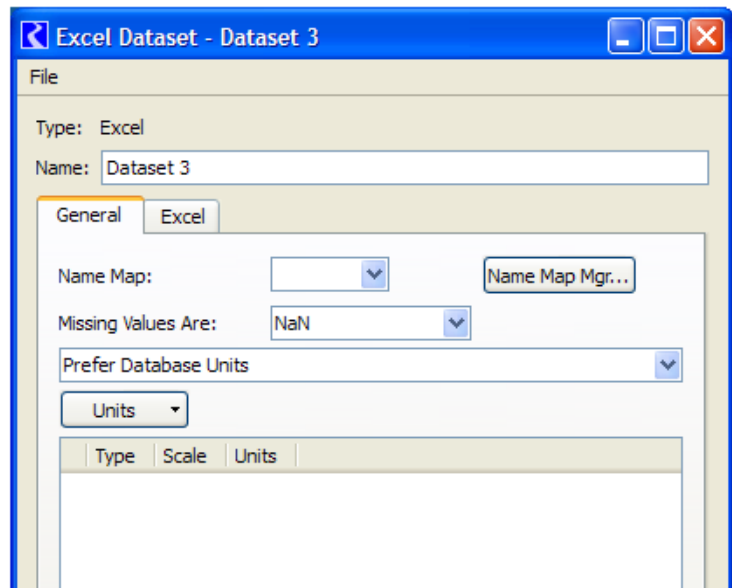
---

**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.

---

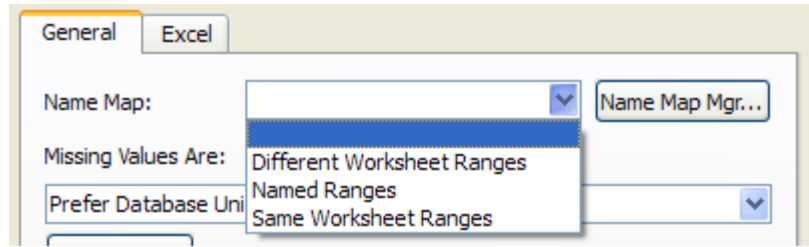
For a use case of creating a DMI to Excel, click [HERE \(Example 2: Excel DMI to export data using Headers\)](#) on page 53.

Double clicking on 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.



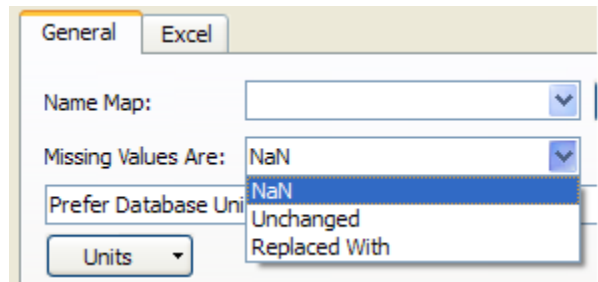
**Name Map:** The user selects the desired Name Map from the **Name Map:** drop down menu as shown in the adjacent screenshot. The selections are populated based on the Name Maps configured in the **Name Map Manager**. For more information on creating Name Maps click [HERE](#)

(page 81). Name maps have special relevance for moving data via range specifications as discussed [HERE](#) (page 74) and for header specification [HERE](#) (page 77)



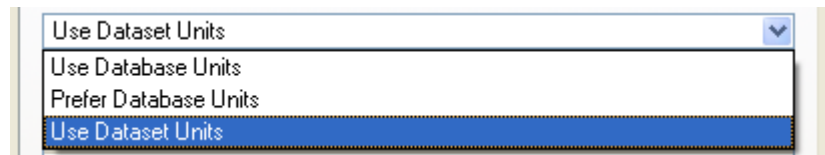
**Missing Values:** The user specifies how missing values are handled with Excel. The current choices are

- **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.** A screenshot is shown in the following figure.



- **Use Database Units:** Since Excel contains no units associated with a value in a cell, the slots' units 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 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

To specify the units to use, follow the same procedure presented [HERE \(Units:\)](#) on page 60.

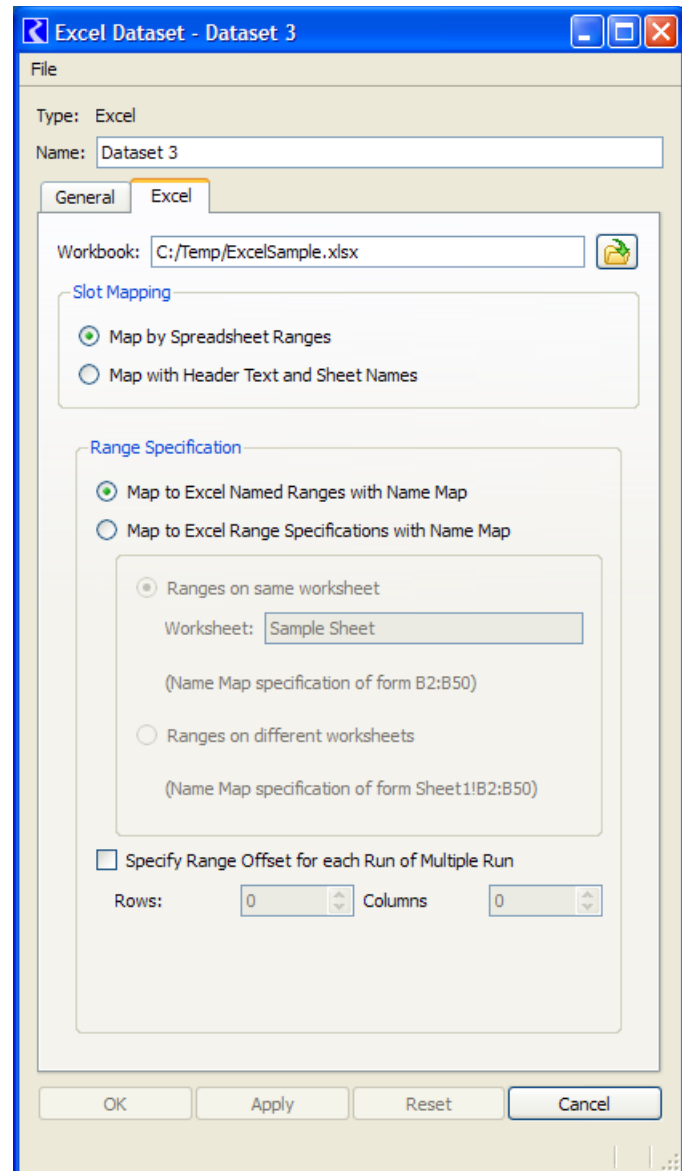
**Excel Tab:** Configuration specific to Excel is specified in the Excel tab of the Excel Dataset editor

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 text box, or the adjacent button can be clicked 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.

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

Following is a table comparing these approaches. Then more information is presented on the configuration options for both the Range and Header approach.



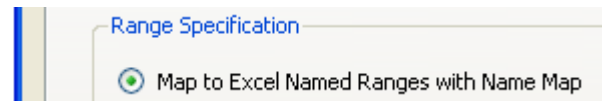
### 4.3.3.1 Table Comparing Excel Dataset Approaches

	Approach	
	Ranges	Headers
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 Map 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/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).

	Approach	
	Ranges	Headers
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. 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.
Where is more information	Click <a href="#">HERE (Section 4.3.3.2)</a>	Click <a href="#">HERE (Section 4.3.3.3)</a>

### 4.3.3.2 Map by Spreadsheet Range

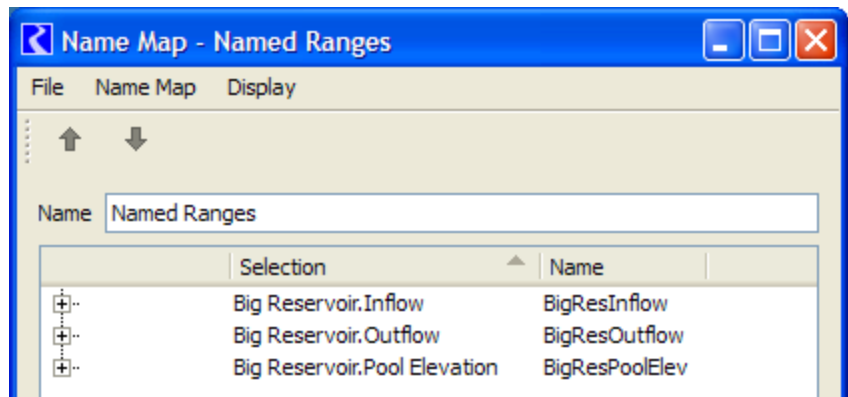
The **Map by Spreadsheet Ranges** approach presents several options for specifying ranges associated with slots. In all cases, the **Name Map** functionality of **Database DMIs** is used to map individual slots to their range specifications.



**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.

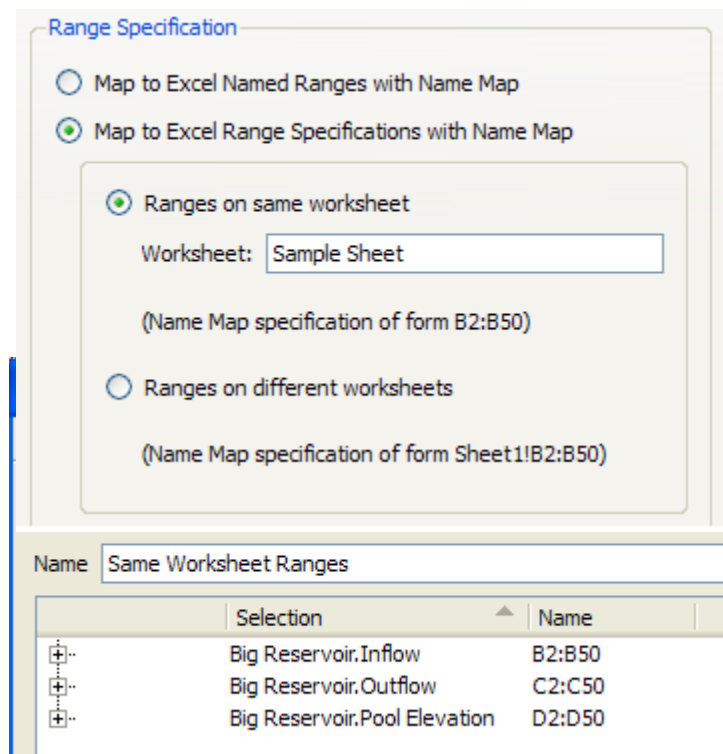
Note that the size of the range specified must match the number of values to be read or written with the DMI. Discussions of the range specification options along with example Name Maps follow.

Under the **Map to Excel Named Ranges with Name Map** option as selected in the screenshot above, 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, BigResOutflow, and BigResPoolElev are named ranges in the Excel sheet.

In the **Map to Excel Range Specifications with Name Map** option, there are two ways to specify ranges for slots. If all of 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. An example of the type of **Name Map** created and selected on the **General** tab of the dataset dialog for this approach is shown to the right.



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. An example of this type of **Name Map** is shown to the right.

**Range Specification**

Map to Excel Named Ranges with Name Map  
 Map to Excel Range Specifications with Name Map

Ranges on same worksheet  
 Worksheet:   
 (Name Map specification of form B2:B50)

Ranges on different worksheets  
 (Name Map specification of form Sheet1!B2:B50)

---

Name:

	Selection	Name
+	Big Reservoir.Inflow	Inflow!B2:B50
+	Big Reservoir.Outflow	Outflow!B2:B50
+	Big Reservoir.Pool Elevation	Elevation!B2:B50

Specify Range Offset for each Run of Multiple Run  
 Rows:  Columns:

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.

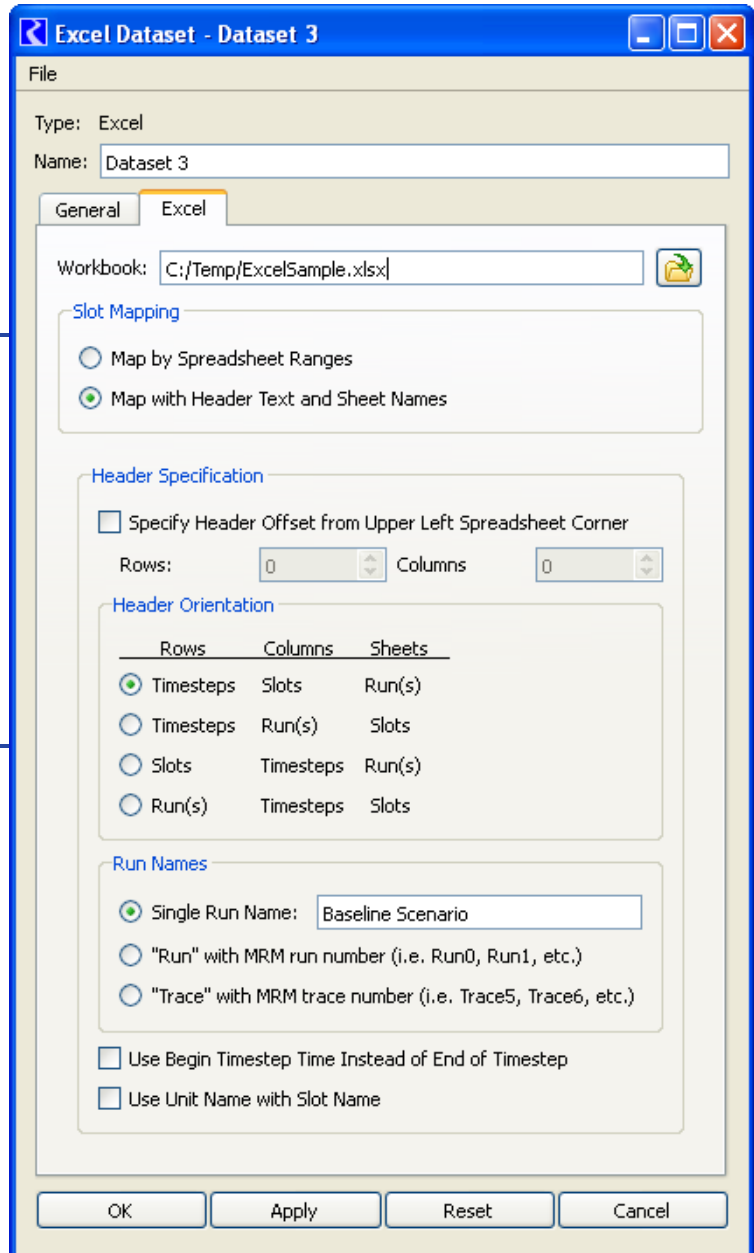
### 4.3.3.3 Map by Header Text and Sheet Names

The **Map by Header Text and Sheet Name** approach is the alternative to using ranges to move RiverWare slot data to Excel. The approach uses a row header, a column header, and a worksheet name to map Excel data with a RiverWare slot. An example of the dataset dialog with this configuration selected appears to the right.

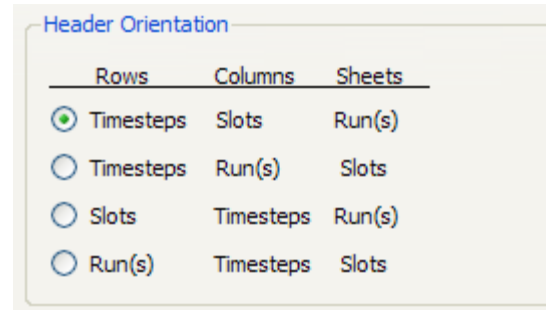
**Tip:** When setting up an input DMI using this Excel dataset approach (especially the first time), it is often easiest to configure the DMI as an **Output** DMI first, run the DMI to create an Excel spreadsheet with the right formatting and desired structure. Edit that spreadsheet or use it as a guide 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.



**Configuration options:** 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 that orientations for putting each timestep on a separate sheet are not supported.



A screenshot of an Excel spreadsheet with the first orientation is shown.

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.

	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

**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 StorageReservoir.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.

The **Run Names** frame indicates how runs in the Excel workbook will be labeled. 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). For multiple runs, a single run name is not adequate

to label each run individually. Two options are provided for labeling multiple runs. If the **“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 that the index number used is zero-based to be compatible with how multiple run data has historically been written out of the ExcelWriter tool. If the **“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. Setting up trace numbers in MRM is discussed [HERE \(Input DMI Runs\)](#).

---

**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.

---

An option is provided with the header approach to **Use Begin Timestep Time Instead of End of Timestep**. Timesteps in RiverWare are dated with the end of timestep time, that is, a daily 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 the **Use Begin Timestep Time Instead of End of Timestep** option 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 daily 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.

# Name Mapping

## 4.4 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 DSS’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 that if the names are mapped within the database itself, as they are in HDB, a Name Map is not needed.

---

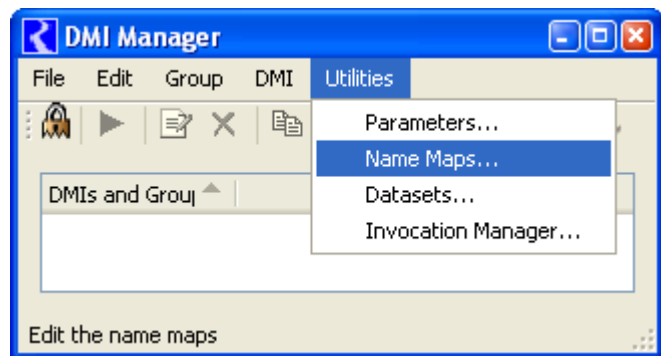
**Note:** There is no need to create a name map if the map is strictly to specify a different case. This will be taken care of automatically. For example, a name map is not necessary for a reservoir in RiverWare called BigLake while in the DSS database, it is called BIGLAKE.

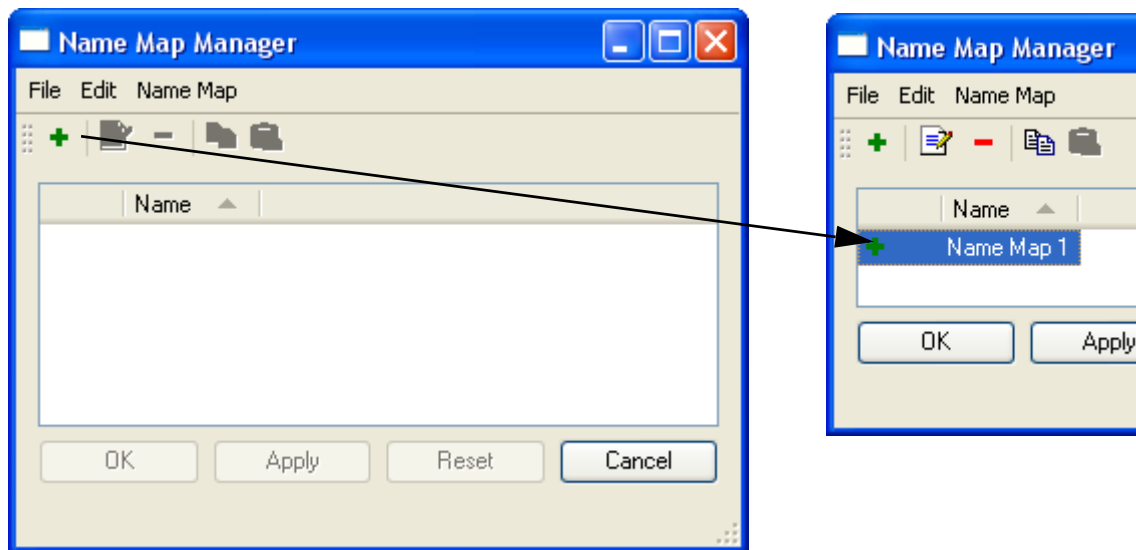
---

### 4.4.1 Configuring Name Mapping

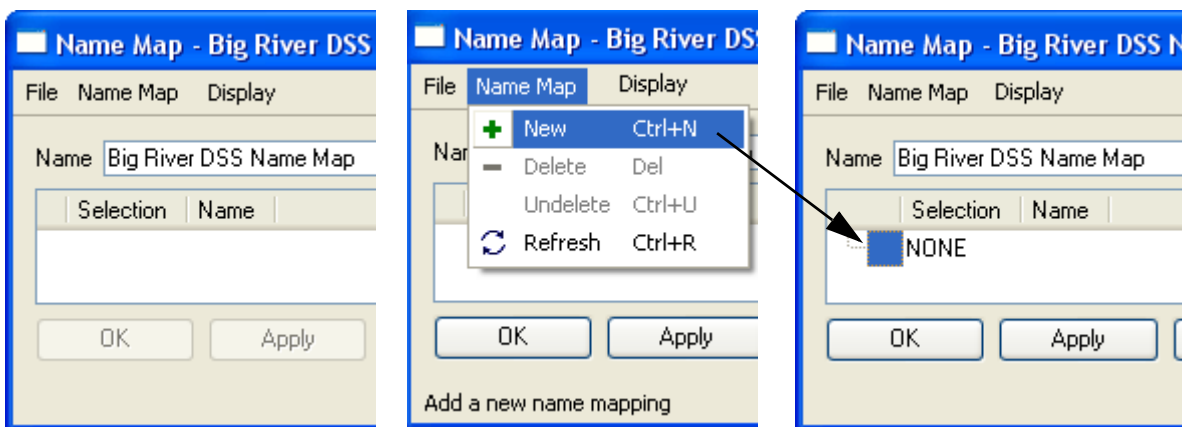
To bring up the name mapping editor from the **DMI Manager**, the user selects **Utilities** → **Name Maps...** as shown in the following screenshot.

This opens a **Name Map Manager** dialog as shown in the following screenshot. Clicking on the **+** icon adds a new Name Map to the editor and gives it a default name.



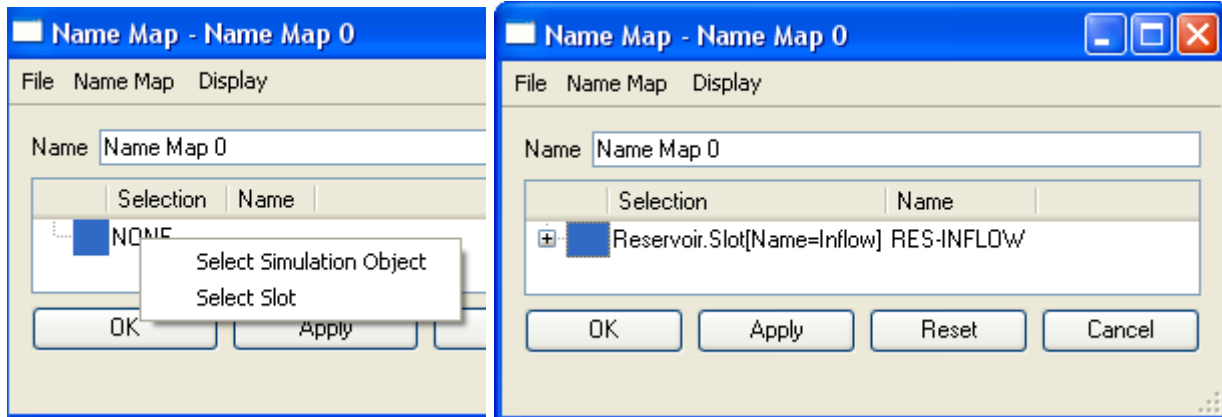


Double clicking on the newly added Name Map opens the **Name Map** dialog as shown in the following screenshot. In this dialog, the user is able to edit the name of the Name Map. Selecting **Name Map** → **New** adds a new entry to the list.

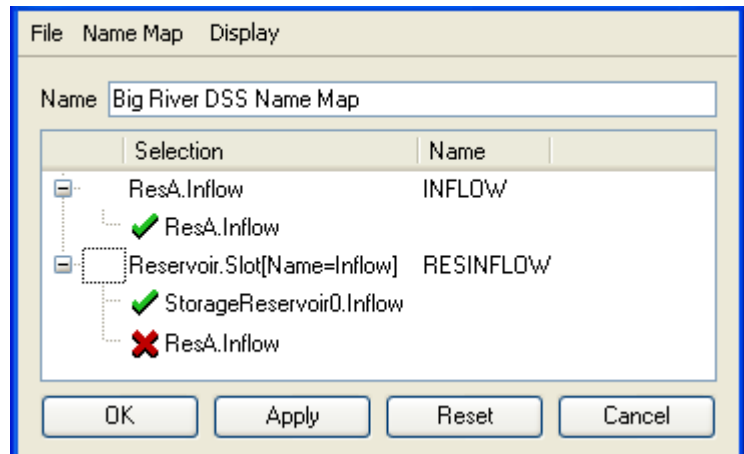


Clicking on the newly selected entry opens the context menu options shown. Either click **Select Simulation Object** or **Select Slot**. This action opens the selector dialog for the appropriate selection. In this dialog, the user selects the desired objects or slots that will be part of this Name Map. 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. More information on using the selector is given [HERE \(Selector.pdf, Section 7.3\)](#). 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, click on 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. An example of this is shown in the screenshot below right.



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 the following screenshot, 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.



#### 4.4.2 Re-ordering entries in the Name Map

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 is click on an entry (Selection and Name row) and then use the up and down arrows   or **Display** → **Move Up** or **Display** → **Move Down** file choices to re-arrange rows. To return to prioritized mode, the user selects **Display** → **Priority**. Regardless of the ordering, the name maps are stored according to priority; custom order is just for display.

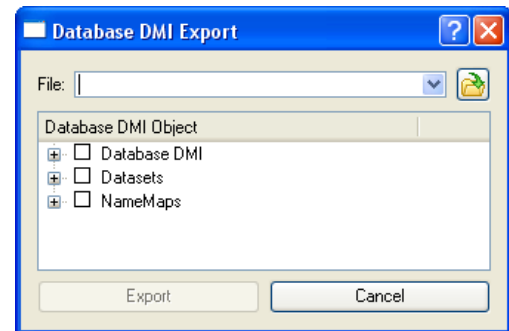
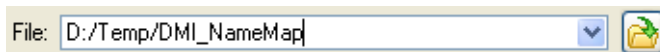
## 4.5 Sharing Database DMI's

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 DMI's, Datasets, and Name Maps. Following are instructions and descriptions of both exporting and importing Database DMI configurations.

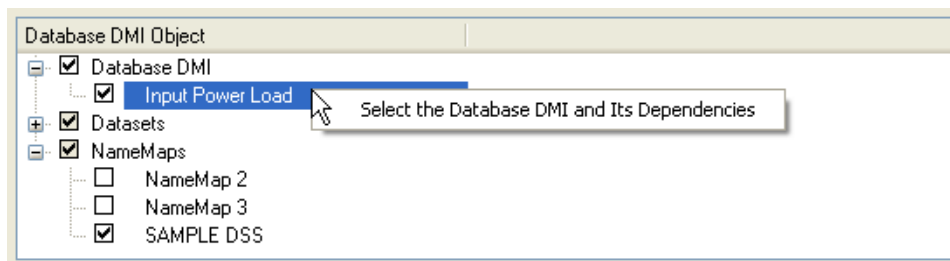
### 4.5.1 Export

To export the Database DMI configuration, the user selects **File → Export Database DMI** from the **DMI Manager**. The following window opens:

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



The user clicks on the  symbol to expand the tree view. Clicking on the check boxes selects the Database DMI, Dataset or NameMap to export. Clicking on 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 the following screenshot. This allows the user to select not only the Database DMI, but also the Dataset and/or NameMap associated with the selection.




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

## 4.5.2 Import

Importing a set of configurations is similar to exporting. The user clicks the **File** ➔ **Import Database DMI** menu on the DMI editor to open the following dialog:



The user types in the file name or uses the browser to select the file to import. Clicking on the **Preview** button reads the file and displays the items in the file. Clicking on 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. If the toggle is off 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, clicking **Import** brings the items into the model.

