

Data Management Interface (DMI) Enhancements

Overview

Bill Oakley

DMI Enhancements

- New features
 - Solicited from users
- Internal changes
 - Determined by developers
 - Motivated by new features
 - Improve code (extensibility, maintainability, performance)

DMI Enhancements

- Import data at any date within a timeseries (available in 4.4)
- Allow a new keyword, *data_date*, in the DMI control file:

`data_date: 1996-01-09 12:00`

- *data_date* must be within the timeseries start and end dates

DMI Enhancements

- Import data from multiple data files from a single control file entry (available in 4.4)
 - Allow multiple *file=<file>* keywords
 - Imported from left to right
 - All data files (except perhaps one) should use the new *data_date* keyword
 - All data files must be in the same scale and units

DMI Enhancements

- Improved handling of *timestep* keyword (available in 4.4)
- Silently ignored unless its value differs from the timeseries value, in which case an error is reported

DMI Enhancements

- Portable DMI definitions (*available in 4.3*)
 - Allow environment variables in control file path and executable path in DMI definition
 - Allow environment variables in control file entries
 - Convert file paths (e.g. '/' to '\') as necessary
 - Windows support for non-native executables (Perl and Tcl)

DMI Enhancements

- Select user-defined parameters in batch mode
- Extend Rcl syntax to include user-defined parameters:

```
InvokeDMI <DMI name> keyword=value  
InvokeDMI <DMI name> !keyword
```

DMI Enhancements

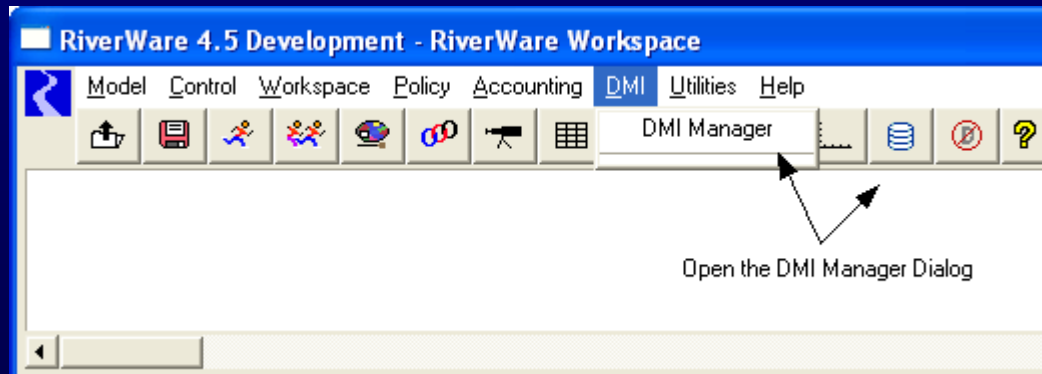
- Complete wildcarding in control files
 - Current wildcarding is limited to:
 - ✓ Concrete simulation object types (e.g. LevelPowerReservoir)
 - ✓ Slot types (e.g. SeriesSlot)
 - Add wildcarding for:
 - ✓ Abstract simulation object types (e.g. Reservoir)
 - ✓ Account types (e.g. StorageAccount)
 - ✓ Account attributes (e.g. water type)

DMI Enhancements

- Complete wildcarding in control files (cont)
 - Add wildcarding for:
 - ✓ Supplies
 - ✓ Supply attributes (e.g. release type)
 - ✓ Exchanges
 - ✓ Paybacks

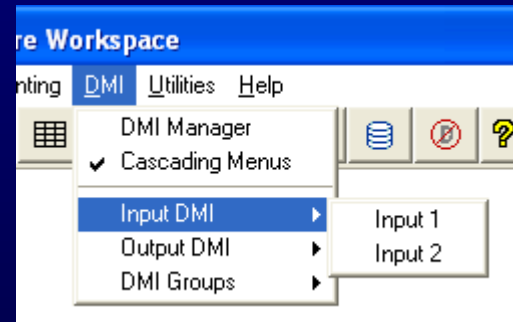
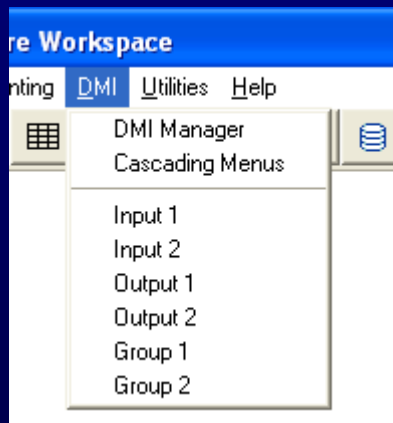
DMI Enhancements

- Workspace DMI quick-pick button (available in 4.4)
- Move DMI from “Utilities” menu to its own menu:



DMI Enhancements

- Workspace DMI quick-pick button (cont)
 - The “DMI” menu will contain a “Manager” menu item and per-DMI and per-group menu items:



DMI Enhancements

- Named DMI groups (available in 4.4)
 - Ability to define a group of DMIs, to be executed in order
 - *Ability to “continue on error”*
 - Ability to determine DMI’s group membership
 - *Nested – groups can contain other groups*

DMI Enhancements

- Graphical user interface for user-defined parameters (available in 4.4)
 - Ability to define parameters – name, type and valid values
 - ✓ Boolean (or flag)
 - ✓ Integer, integer range, *integer list*
 - ✓ Double, *double list*
 - ✓ String, *string list*
 - ✓ “Special” types (e.g. month, day of month)

DMI Enhancements

- Graphical user interface for user-defined parameters (cont)
 - Ability to select parameters – name and value
 - Ability to export/import (share) parameter definitions
 - *“System” parameters (e.g. model, controller, ruleset, accounting period)*

DMI Enhancements

- Graphical user interface for creating control files
 - Ability to select any slot in the model (the "Grand Unified Slot Selector")
 - Ability to select slots using slot types and names, object types and names, and (where appropriate) object attributes
 - Ability to select the slots now (fully specified slot names) or in the future (wildcarding)

DMI Enhancements

- Graphical user interface for creating control files (cont)
 - Example – select “all reservoir inflow slots”
 - ✓ Now – fully specified slot names
 - BlueMesa.Inflow:
 - Crystal.Inflow:
 - ...
 - TaylorPark.Inflow:
 - ✓ In the future – wildcarding
 - Reservoir.Inflow:

DMI Enhancements

- Graphical user interface for creating control files (cont)
- Ability to specify slot keywords:

Reservoir.Inflow: file=... scale=... units=...

DMI Enhancements

- Improved control file parsing (internal)
 - Current parsing is “manual” if-then-else parsing; difficult to extend
 - New parsing will be Flex/Bison based

DMI Enhancements

- General improvements (internal)
 - Remove data structures and methods made obsolete by development which postdates the original DMI development