



Technical Documentation Version 5.1

Batch Mode and RCL



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.

Batch Mode and RCL Table of Contents

Batch Mode and RCL	1
Introduction	1
Running RiverWare from the Command Line	1
Running RiverWare in Batch Mode	2
RCL: The RiverWare Command Language	2
Key	2
Special Parameters	3
Syntax	3

Batch Mode and RCL

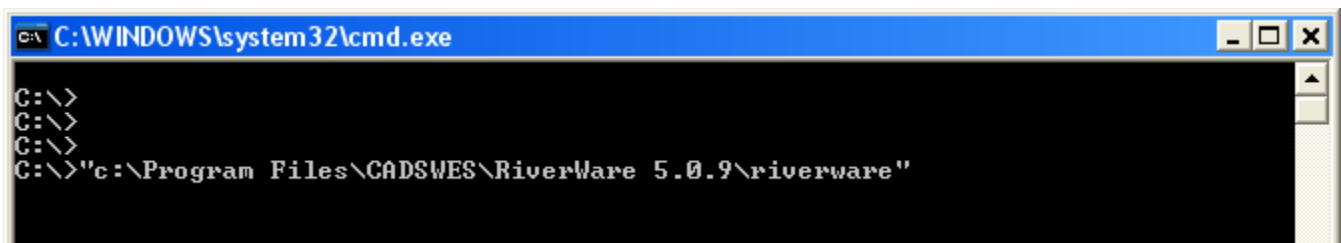
1. Introduction

The RiverWare Command Language (RCL) facility allows models to be run in batch mode. While in this mode, RiverWare does not open a user interface, but instead runs in background mode.

In batch mode, the user provides RiverWare with a script file, which contains the commands RiverWare executes. This means, essentially, that commands are entered through the script file, rather than through the user interface. RCL defines the commands which the script file may contain. This is described in detail in the following document.

2. Running RiverWare from the Command Line

On Windows, the user typically starts RiverWare by clicking on an icon or using the start menu. RiverWare supports loading models and rulesets and lots of other operations from the command line. To access the command line on windows, click the Start Menu -> Run. Then type in `cmd`. A window similar to the following opens. To start RiverWare from this window, you need to type in the location of RiverWare (depending on your PATH configuration)



```
C:\WINDOWS\system32\cmd.exe
C:\>
C:\>
C:\>
C:\>"c:\Program Files\CADSWES\RiverWare 5.0.9\riverware"
```

A complete list of command line options can be viewed by invoking Riverware with the `--help` command line argument. On Windows, in a command line window (`cmd`), this looks like this:

```
riverware.exe --help
```

On the Windows, when RiverWare is executed from a command line window (batch mode or interactively), it is run as an independent process from the command shell, so exiting the command window will not automatically terminate RiverWare. The Windows task manager can be used to terminate batch runs of RiverWare.

Sometimes, the user wishes to run one or more models as automatic tasks (either “cronjobs” on solaris or “Scheduled Tasks” on windows). Setting up the runs using batch mode and the command line

options can be useful in this approach. For example, a user may wish to run a series of models overnight when the machine is more available. One command line option that is useful is the `--log <file>` switch. This option sends any diagnostic output to the specified file. On widows, this is especially useful because without this switch, a console window opens and the user has to “Press any key to exit...” stopping any additional models from running. The `--log` option sends this output to a file and prevents the command prompt window from opening.

3. Running RiverWare in Batch Mode

The syntax to invoke RiverWare in batch mode from a command prompt (either a command window in windows or a terminal window on solaris) is: `riverware --batch <script file>`

The `<script file>` defines how the batch run should proceed. In the file, you specify commands to load a model, load a ruleset if necessary, start a run, and save the results. In the simplest case, you will load a model, run a simulation, save the model, and exit batch mode. The following RiverWare Command Language commands support these actions:

```
// Load the model
OpenWorkspace <Model Name>
// run the simulation:
StartController
//save the model (in your home directory):
SaveWorkspace <Model Name>.saved.mdl
// Close the opened model and exit RiverWare:
CloseWorkspace
```

This is a very simple example. The full complement of key words and commands is presented in the following section.

4. RCL: The RiverWare Command Language

4.1 Key

- Angled brackets (`<>`) indicate a required parameter.
- Brackets (`[]`) indicate an optional parameter.
- A vertical bar “|” within angled brackets or brackets indicates a choice of parameters. For example, “`<#RunInfo|#MRM>`” indicates that either “`#RunInfo`” or “`#MRM`” must be provided as a required parameter.
- Parameters starting with “!” are tokens which should appear exactly as is.

- Parameters starting with a “#” are RiverWare names which should appear exactly as is.
- A “#” sign at the beginning of the line represents a comment. Anything occurring after the “#” will be ignored.
- Parameters which contain embedded white space characters must be enclosed in braces [{}].

4.2 Special Parameters

•{date time}

A date and time of the format {month/day/year hour:minute}, where the year includes the century and the hour is 24-hour based. Note that a date and time, which includes embedded white space, must be enclosed in braces.

•{delta time}

A timestep or time duration of the format {count units}, where units are “HOURS,” “DAYS,” “WEEKS,” and “MONTHS.” Note that a delta time, which includes embedded white space, must be enclosed in braces.

4.3 Syntax

•SetTrace <1|0>

Enable or disable trace messages from the RCL interpreter. The trace messages are prefaced with “TRACE:” and written to standard output; they can be captured in a file with the “--log <log file>” command-line option.

•SetDiagFile <diagnostic file>

Sets the location of the diagnostic output file.

- <diagnostic file> is the path to where the diagnostic file will be written.

•OpenWorkspace <file path>

Loads a model into the workspace.

- <file path> is the path to where the model resides on disk.

•SaveWorkspace [file path]

Saves the model which is currently loaded into the workspace.

- [file path] is the path to where the model is to be saved on disk. If the file path is not specified, the OpenWorkspace file path is used.

•CloseWorkspace

Clears the workspace.

•SyncObj [!Acct] [!ExDiffTS]
[!StartDate <start date>]
[!EndDate <end date>]

Synchronize slots to either the run control or the specified dates.

- [!Acct] includes accounting slots.
- [!ExDiffTS] excludes slots whose timestep differs from the run control timestep.
- [!StartDate <start date>] synchronizes slots to the specified start date, rather than the run control initial date.
- [!EndDate <end date>] synchronizes slots to the specified end date, rather than the run control end date.

•SlotList <output file>

Writes information about all slots in the model to the output file.

- <output file> is the file the slot list is written to.

The information is written as comma-separated values to the output file; the “--help” command line option lists the output fields.

•StartController [!MRM <MRM configuration>]

Starts the current RiverWare controller.

- !MRM will start the Multiple Run Management controller instead of the single run controller, using the specified MRM configuration. RCL currently does not support commands for defining multiple runs, so models should already contain the multiple run definitions.

•InvokeDMI <DMI name> [-UserParam=value]

Invokes a DMI.

- <DMI name> is the name of the DMI to invoke. If the name contains embedded white space, it must be enclosed in braces.
- [-UserParam=value] sets the user parameter “userParam” to “value”.

•InvokeDssDMI <DMI name> <CWMS file> <CWMS F part>

Invokes a DSS DMI within the Corps of Engineers Corps Water Management System (CWMS).

- <DMI name> is the name of the DSS DMI to invoke. If the name contains embedded white space, it must be enclosed in braces.
- <CWMS file> is the name of the CWMS DSS file. If the name contains embedded white space, it must be enclosed in braces.
- <CWMS F part> is the CWMS F part.

•ListDMI [!Type <Input|Output>]
[!Dataset <dataset type>]<output file>

Writes the names of the DMIs which match the specified criteria to the output file.

- [!Type <Input|Output>] limits the output to either Input or Output DMIs.

- [!Dataset <dataset type>] limits the output to DMIs which use only the specified dataset type.
- <output file> is the file the DMI list is written to.

For example:

- ListDMI file

Writes the names of all DMIs to the file.

- ListDMI !Type Input file

Write the names of all Input DMIs to the file.

- ListDMI !Type Input !Dataset DSS file

Write the names of all Input DMIs which use only DSS datasets to the file.

•SlotListDMI <DMI name> <output file>

Writes information about the slots to/from which the DMI imports/exports data; the information is written as comma-separated values to the output file. The information includes:

- The slot's name.
- The slot's priority (determined by its dataset association).
- The slot's begin date "mm-dd-yyyy".
- The slot's begin time "hh:mm:ss".
- The slot's end date "mm-dd-yyyy".
- The slot's end time "hh:mm:ss".
- The dataset associated with the slot .
- The dataset's type, currently DSS or HDB. If the Dataset is DSS, then the slot's DSS path "/A/B/C/D/E/F/" is also provided as another comma separated value.

•GetRunInfo <#RunInfo|#MRM>

```
[!InitDate] [!EndDate] [!Duration] [!Step]
[!Controller]
```

Prints single run information (#RunInfo) or multiple run information (#MRM) to standard output.

- !InitDate prints the run's initial date.
- !EndDate prints the run's end date.
- !Duration prints the run's duration.
- !Step prints the run's timestep.
- !Controller prints the current controller.

If no parameters are specified, all information is printed.

•SetRunInfo <#RunInfo|#MRM>

```
[!InitDate <date time>] [!EndDate <date time>]
[!Duration <delta time>]
```

[!Controller <controller type>]

Sets single run information (#RunInfo) or multiple run information (#MRM).

- !InitDate sets the run's initial date to <date time>.
- !EndDate sets the run's end date to <date time>.
- !Duration sets the run's duration to <delta time>.
- !Controller sets the run's controller to <controller type>. The controller type must be one of the controllers available in the Run Control dialog:

Simulation

Rulebased Simulation

Note that controller types which contain embedded white space must be enclosed in braces.

•GetSlot <slot name> <date time>

Returns the slot's value at the specified timestep as a Tcl variable. The command is used when writing batch mode scripts which make use of the Tcl scripting language, which is beyond the scope of this reference. Please contact RiverWare support for more information.

•SetSlot <slot name> <date time> <value>

Sets the slot's value at the specified timestep to the value, assumed to be in standard scale and units.

•LoadConstraints <file path> [!Append]

Loads the constraint set into the workspace. By default, constraint sets which are already loaded into the workspace are cleared.

- <file path> is the path to where the constraint set resides on disk.
- !Append does not clear constraint sets which are already loaded into the workspace.

•LoadOptSet <file path>

Loads the optimization ruleset into the workspace.

- <file path> is the path to where the optimization ruleset resides on disk.

•LoadRules <file path>

Loads a ruleset into the workspace.

- <file path> is the path to where the ruleset resides on disk.

•OpenGlobalSet <file path>

Opens a global function set. Note, this must be called before LoadRules if the ruleset references functions in the global set.

- <file path> is the path to where the set resides on disk.

•Output <outputName>

Generates an output file. The output file must already be specified in the Output Manager of the model. Plot pages cannot be generated. If an Xmgr Data Format File is to be generated, the option Save to File (rather than send to Plot Program) must be selected in the Output Manager.

- <outputName> is the name of the output file saved in the Output Manager of the model.