6. Distributed Concurrent Runs

When performing concurrent MRM runs consisting of many runs, the following are some problems that users encounter:

- The memory required may exceed the resources available and/or
- The time required to make the runs is excessive.

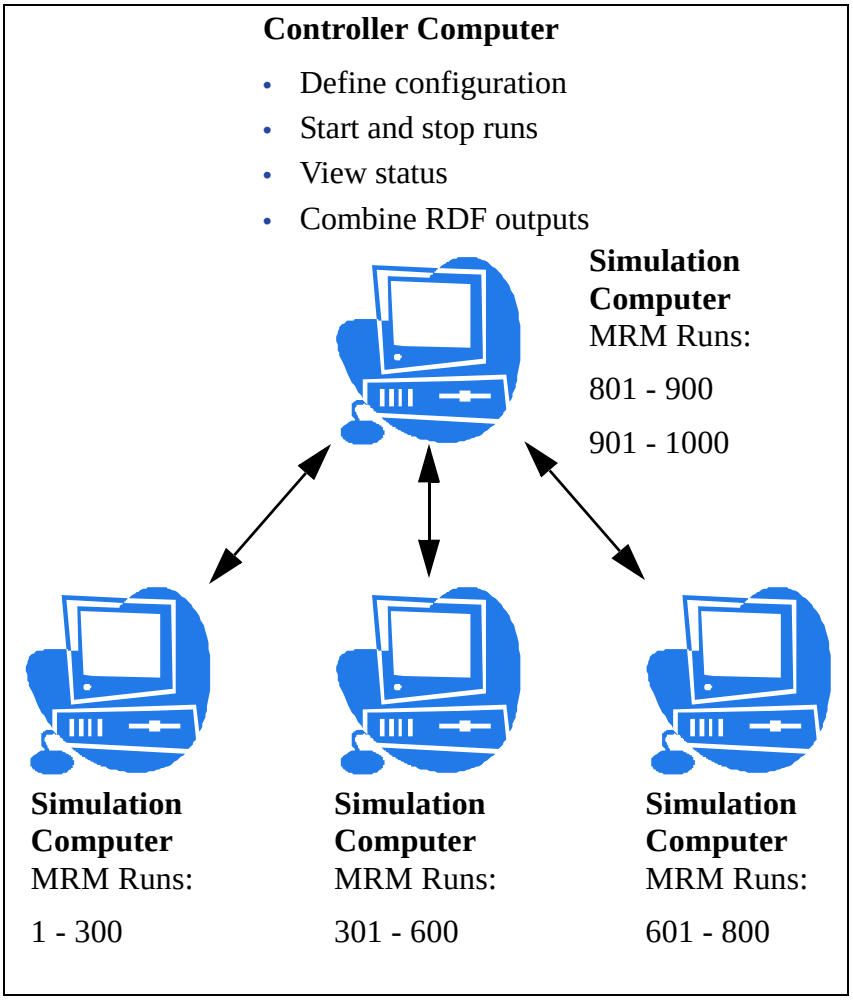
This section describes a utility that solves these two issues by distributing many runs across many computers. In this approach, there is a controller computer and simulation computers. The controller computer (which may also be a simulation computer) does the following:

1. Creates the configurations

2. Controls execution (Start and Stop) on each simulation computer

3. Tracks the progress of runs on the simulation computers

4. When all the runs are finished, combines the output RDF file from each simulation computer into one output RDF file.



Each simulation computer then executes the MRM run(s) that the controller computer gives it to execute. Each simulation computer runs one or more MRM runs that consist of a portion of the total number of concurrent runs. This is shown graphically in the screenshot. In this example, there are 1000 runs that are distributed unequally to four machines. One of the machines is also the controller computer. That machine has 2 CPUs, so it is assigned two MRM runs.

This feature was initially designed for a very specific application of concurrent MRM. As such, the following are the limitations and constraints:

- You can only distribute runs across Windows machines
- The same version of RiverWare must be installed on each machine

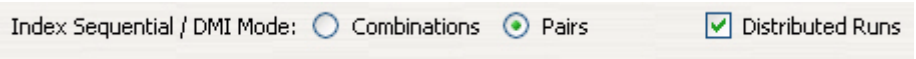
- The MRM configuration must be in **Pairs** mode LINK (this will be relaxed in the future.)
- Each machine can access (read/write) two network directories: the working directory and the model directory
- No per-run output DMI's are allowed

This document is organized to present an overview of the user interface, how to make a run, and how the utility works to distribute the runs.

6.1 User Interface Overview

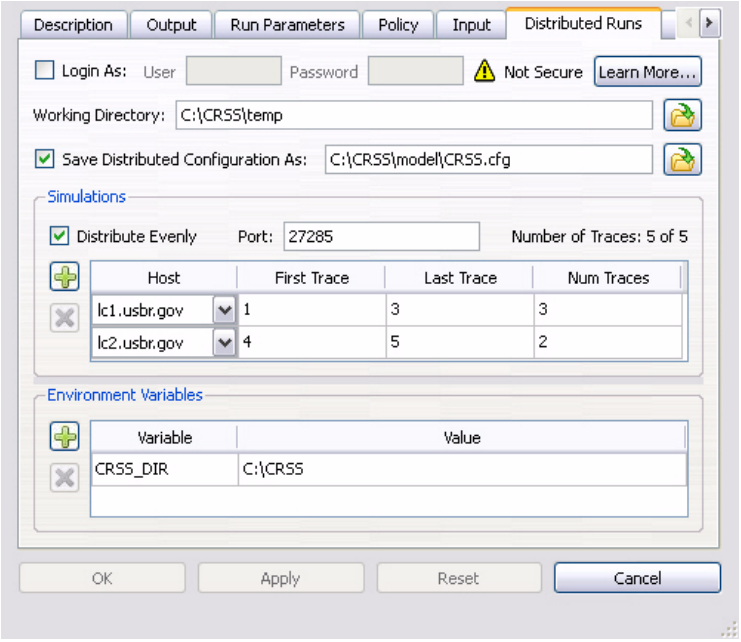
The interface for distributing MRM runs across multiple machines consists of two components, the MRM configuration within RiverWare and the Distributed MRM dialog which is external to RiverWare.

6.1.1 MRM Configuration



Currently, MRM runs can only be distributed when in Pairs mode. TODO link. Thus, when in **Pairs** mode, the **Input** tab has a **Distributed Runs** check box that becomes enabled. When checked, the **Distributed Runs** tab is added to the dialog. As discussed in the sections that follow, the **Distributed Runs** tab allow you to configure:

- Whether a login is necessary, and optionally the user and password.
- The working directory.
- Whether the configuration should be saved to a file.
- The TCP/IP port.
- For each individual simulation, the host address, the first trace and the number of traces.
- Environment variables and their values.



Here's a screenshot of the Distributed MRM tab:

6.1.1.1 Login

If a login is required, check the box to enable the user and password fields. If either the user or password is omitted but required when the model is run, you will be prompted for it.

> **Note:** Although it is not explicitly shown, entering the password here is **not** secure. The password is not encrypted in the model file, so anyone who has access to the model file can find the password. However, when a distributed run is made, the password **is** written across the network using a secure, encrypted TCP socket. Click the **Learn More** button to open a dialog explaining the security issues in detail.

6.1.1.2 Working Directory

A distributed concurrent run creates several "working" files - batch script files, control files, intermediate RDF files and log files among them. The working directory specifies where the working files will be created. Importantly, it should be a network directory which the controller computer and all simulation computers have access to (via the same path).

6.1.1.3 Save Configuration as

When a distributed concurrent run is started, RiverWare writes a configuration file, invokes the Remote Manager process, and exits. If a user elects to save the configuration to a named file, it is possible to invoke the Remote Manager directly bypassing RiverWare.

6.1.1.4 Host Table

The host table defines the simulation computers and the traces they will simulate.

Distribute Evenly: If "Distribute Evenly" is checked the traces are distributed evenly among the simulation computers, and the "First Trace" and "Num Traces" fields aren't editable.

Port: The TCP/IP port the controller computer will use to connect to the simulation computers. It may vary by site based on firewall issues and other installed applications.

Number of Traces: The number of traces assigned to simulation computers. In the above screenshot 400 of 1000 traces have been configured. (The 1000 comes from the "Input" tab.) If the traces are "under configured" or "over configured" the text will be displayed in red.

Host, First Trace and Num Traces: For each simulation computer the user will specify the host name or IP address and the range of traces it will simulate (unless "Distribute Evenly" is checked). The host name or IP address is entered or selected using an editable combo box (left and right, respectively):

RiverWare will remember previously entered host names and IP addresses, simplifying configuration. There will be a menu item to clear the host list, for example prior to distributing the model.

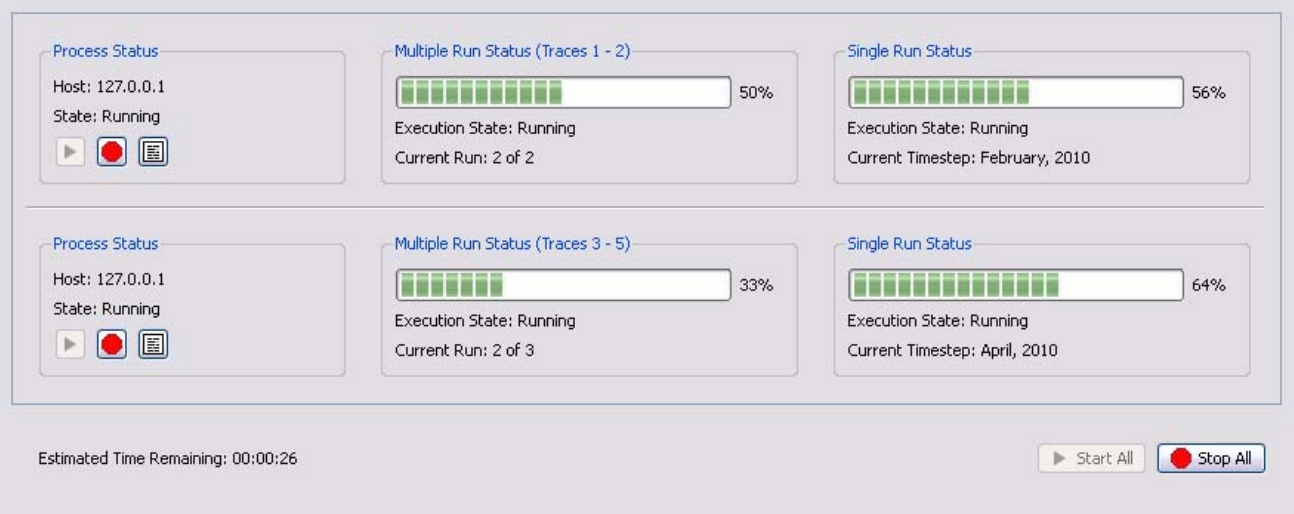
6.1.1.5 Environment Variables

You can enter environment variables and value pairs.

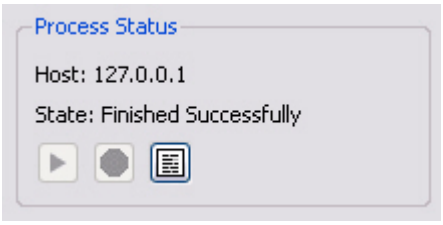
6.1.2 Remote Manager and Status dialog

As mentioned above, the Remote Manager includes a user interface which displays the status of the simulations. This dialog is not within RiverWare but is a separate executable in the installation directory. It is also opened automatically when you make a distributed MRM run from the RiverWare MRM Run Control. More specifically, the Remote Manager user interface allows you to:

- Start and stop the simulations individually or collectively.
- View RiverWare's diagnostic output for the simulations.
- View the multiple and single run status.
- See an estimated time remaining.
- See the status of the post-processing (combining the RDF files).

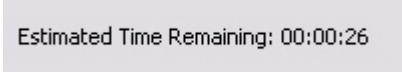


From left to right are the "Process Status" panels, the "Multiple Run Status" panels and the "Single Run Status" panels. In better renditions, here is the "Process Status" panel with the "start", "stop" and "view diagnostics" buttons. The buttons allow you to start, stop, and view diagnostics for the individual run/process.



The “Multiple Run Status” and “Single Run Status” panels (which are very similar to RiverWare’s run status dialog) display the progress of each MRM run and each individual run:

The bottom of the dialog shows the estimated time remaining based on available data after the first run has completed.



6.2 Setting up the machines

Before making a run, you must set up each machine. This section assumes you have installed RiverWare on each machine and it is located in “C:\Program Files\CADSWES\RiverWare 5.2”. Following are instructions to set up each machine for the distributed run. For now, the installation is performed manually and requires administrator permissions. There are four new executables which are co-located with riverware.exe:

- RwService.exe
- RwSvcCtlr.exe
- RwRemoteMgr.exe
- CombineRdf.pl

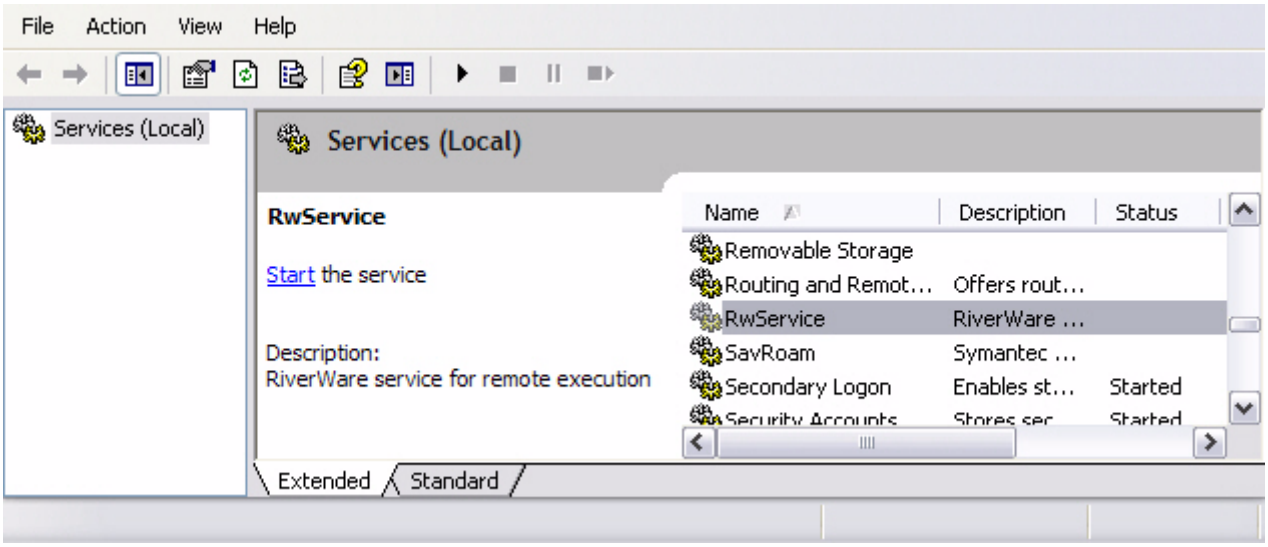
All must be present but for the most part, you will only interact with RwRemoteMgr.exe and RwService.exe.

6.2.1 Installing RwService.exe

The RiverWare service, RwService.exe, must be running on each simulation computer. The RiverWare Service Controller, RwSvcCtlr.exe, is used to install RwService.exe. From a Windows Command Prompt:

```
cd C:\Program Files\CADSWES\RiverWare 5.2
RwSvcCtlr.exe -i “C:\Program Files\CADSWES\RiverWare 5.2\RwService.exe”
```

Once the service has been installed it can be started using the Windows Service dialog: TODO HOW DO YOU GET THIS?



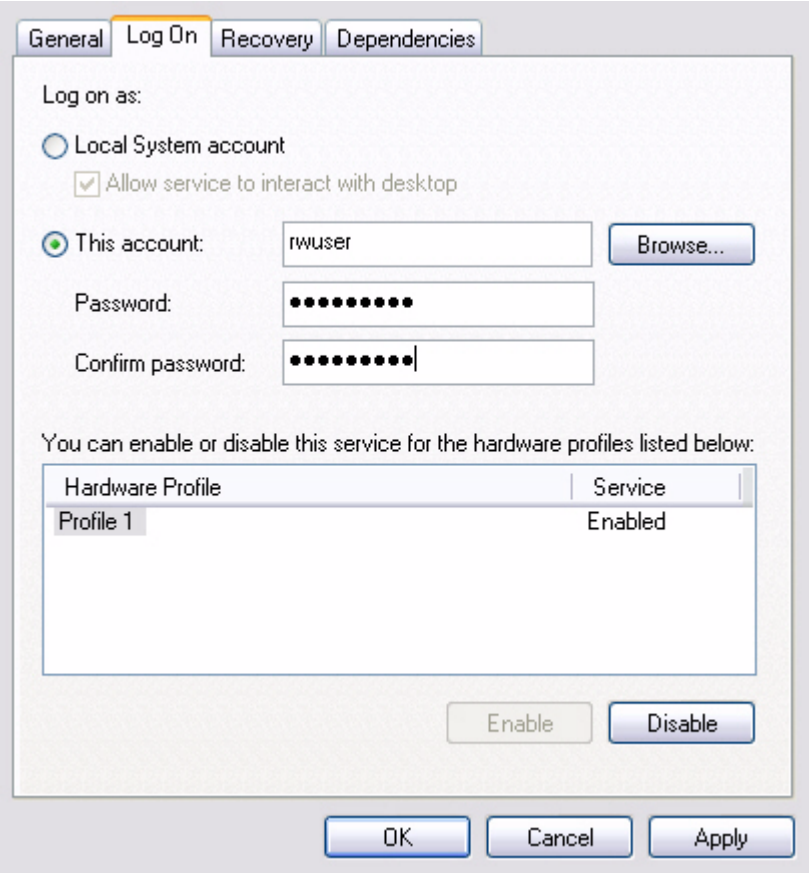
Although not shown, the service can be stopped, paused and restarted using the Service dialog (although it must be started and not paused for the distributed simulations to run).

RwSvcCtlr.exe is also used to uninstall the service:

```
cd C:\Program Files\CADSWES\RiverWare 5.2
RwSvcCtlr.exe -u “C:\Program Files\CADSWES\RiverWare 5.2\RwService.exe”
```

6.2.2 Setting The Service Log On

By default a program started by a service has very limited permissions. RiverWare is started by the RiverWare service on the simulation computers, so by default it has very limited permissions. For example, it might not be able to access network directories or write in certain directories. To alleviate this, it might be necessary for the RiverWare service to run as a specific user. To set this up, right-mouse-click on the "RwService" line in the Service dialog, select "Properties" and select the "Log On" tab. Here you can specify the user the RiverWare service will run as:



Although not shown, in the Properties dialog, in the "General" tab, it's possible to specify whether RwService is started manually (through this dialog, the default) or automatically (when the computer starts up).

6.3 How to Make a Distributed Run

You make a Distributed MRM run by starting execution from the controller computer. There are two options, creating/changing the configuration (within RiverWare) or re-running a configuration(can be external to RiverWare). These are described in the next two sections:

6.3.1 Creating or changing a configuration

If you are creating a new configuration or changing an existing configuration, the changes must be made from within RiverWare. This will allow RiverWare to create the necessary configuration files that will be passed to the simulation controllers. When you click start, RiverWare will create the necessary configuration files, start the RiverWare Remote Manager, and then exit. The RiverWare Remote Manager controls the execution of the MRM runs on the simulation computers. Here are the steps to making the runs in this case:

1. Open RiverWare

2. Fully define the configuration or make any changes to an existing configuration in the MRM Run Control Configuration dialog. Click TODO LINK for the options.

3. Apply the changes.

4. SAVE THE MODEL. The distributed runs open and run the model that is saved on the file system. Therefore, you should save the model now, so that the configuration is preserved.

5. Click Start on the Multiple Run Control Dialog. RiverWare will start the Remote Manager and then start the shutdown sequence. It will prompt you for confirmation so you can cancel at any time.

6. From the Remote Manager **HERE (Section 6.1.2)**, click the start button to start the distributed runs.

7. The individual runs start and the status is shown including an estimate of the time remaining.

8. When all runs are complete, the output RDF files are combined into one final RDF file.

6.3.2 Re-running a configuration

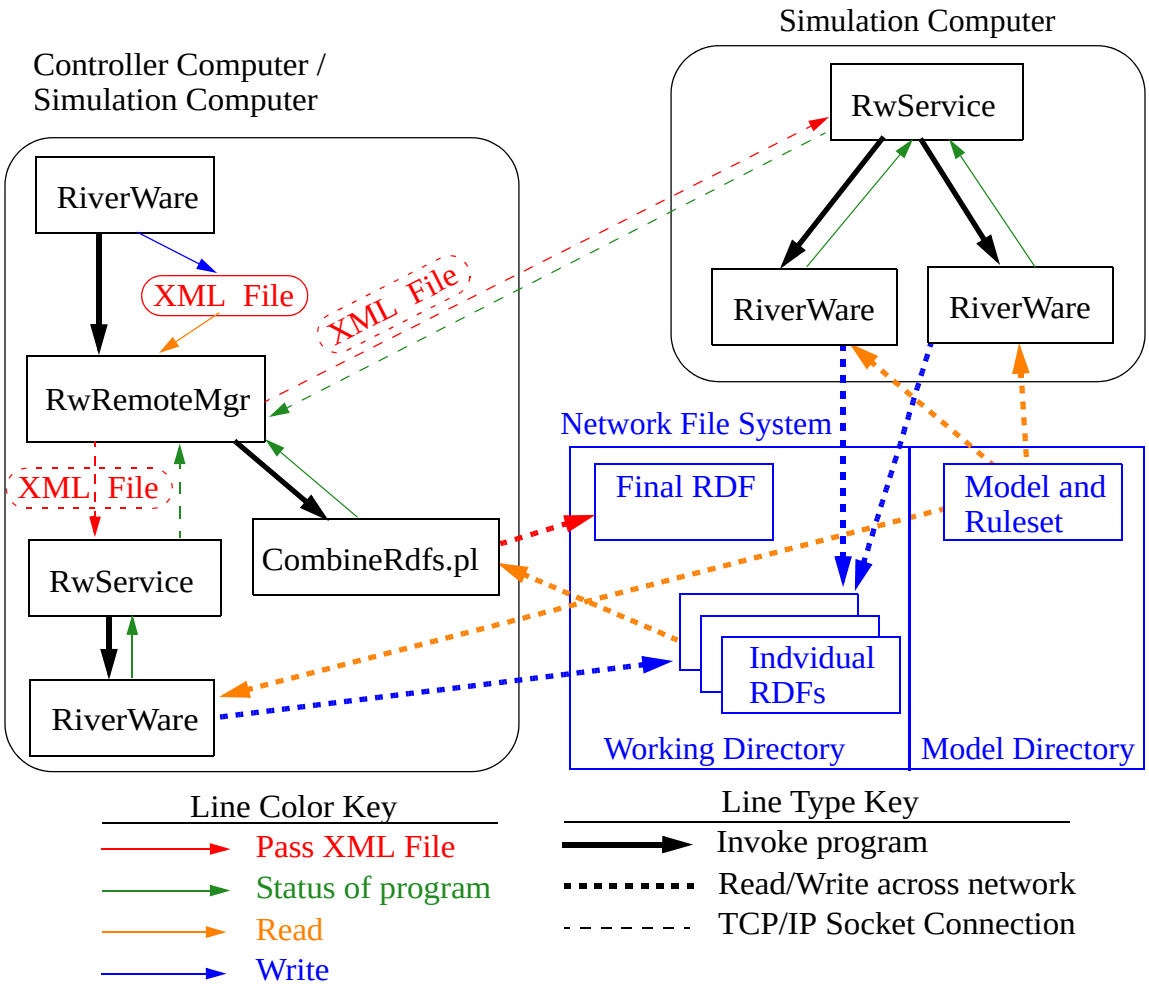
If you are repeating a previously saved configuration, then you can execute the RiverWare Remote Manager directly and not open the RiverWare. In that case, you start at step 6. above.

6.4 How it Works

In the distributed architecture there is a “controller” computer and one or more “simulation” computers; notice that:

- The controller computer can also be a simulation computer.
- A simulation computer with adequate capacity can support multiple simulations.

The distributed architecture introduces three new processes - the RiverWare Remote Manager (RwRemoteMgr), RiverWare Service (RwService), and the RiverWare Service Controller (RwSvcCtlr, not shown).



6.4.1 RiverWare Service

The RiverWare Service runs on each simulation computer; it's a Windows Service which runs in the background, listening to the network for incoming connections (over a TCP/IP socket). Each connection represents a simulation; the Service reads the simulation's XML configuration from the socket, creates a batch script file, and invokes RiverWare in batch mode. The Service reads RiverWare's output and writes it to the socket.

The RiverWare Service is generic - it reads an XML configuration, creates a batch script file and invokes RiverWare in batch mode. The XML configuration can define any batch mode invocation.

6.4.2 RiverWare Service Controller

The RiverWare Service Controller controls the RiverWare Service - it allows a user to install, start, stop, pause, resume, and uninstall the Service. Once installed the Service is a “real” Windows Service and can also be controlled through the Windows Service interface. The Service must be installed and started (and not paused) to receive incoming network connections.

6.4.3 RiverWare Remote Manager

The RiverWare Remote Manager runs on the controller computer. It parses an XML configuration file which defines the simulations and:

- Creates an XML configuration for each of the simulations.
- Configures its user interface (a dialog which shows the status of the simulations).
- For each simulation, connects to the simulation computer (over a TCP/IP socket), writes the XML configuration to the socket, and reads RiverWare’s output from the socket (which it uses to update its status dialog).
- When all simulations have finished, combines the partial RDF files to create the final RDF files.

6.4.4 XML Configurations

Previous sections have referred to XML configurations. These are the files that RiverWare creates to control/define the distributed MRM runs.

> **Note:** This section is a technical reference providing more information on how this utility works. The XML files are generated by the utility and the user does not need to edit these files to make a distributed MRM run.

There are three distinct XML configurations which exist as top-level elements in an XML document; the configurations are identified by their names and can coexist within a document:

- “distrib” identifies a distributed concurrent run (hereafter referred to as the “distributed configuration”).
- “RW” identifies a RiverWare batch mode invocation (hereafter referred to as the “RiverWare configuration”).
- “GenApp” identifies a generic application invocation.

For example, an XML document which defined a distributed concurrent run, two RiverWare batch mode invocations and a generic application invocation would have top-level elements:

```
<document>
  <distrib>
    ...
  </distrbv>
  <RW>
    ...
```

```
 </RW>
 <RW>
  ...
 </RW>
 <GenApp>
  ....
 </GenApp>
</document>
```

The following sections describe the Distributed Configuration, RiverWare Configuration and GenApp.

Distributed Configuration: The distributed configuration document is created by RiverWare when a user starts a distributed concurrent run. DistribMrmCtlr parses the distributed configuration and creates the RiverWare configurations for the simulations. Some elements are from the MRM configuration, the others are provided by RiverWare. A "Sample" of this XML file is shown below with key elements preceded by brief descriptions:

<distrib>

If required, the login information from the MRM configuration. Missing user or password is prompted for.

<login user="" **passwd**=""/>

The RiverWare executable which started the distributed concurrent run. The assumption is that all simulation computers have this executable installed.

<app>C:\Program Files\CADSWES\RiverWare 5.2\riverware.exe**</app>**

The model loaded when the user starts the distributed concurrent run. The assumption is that all simulation computers can access the model using the same path.

<model>R:\\CRSS\\model\\CRSS.mdl**</model>**

The config attribute is the MRM configuration selected when the user starts the distributed concurrent run. The mrm elements are from the MRM configuration and identify the individual simulations - the simulation computer and the traces to simulate. Although the mrm element supports per-simulation-computer port numbers, the assumption is that all simulation computers use the same port number (and the MRM configuration dialog allows a single port number to be entered).

<mrmlist config="Powell Mead 2007 ROD Operations">

<mrm addr="lc1.usbr.gov" **port**="27285" **firstTrace**="1" **numTrace**="200"/>

<mrm addr="lc2.usbr.gov" **port**="27285" **firstTrace**="201" **numTrace**="300"/>

</mrmlist>

The rdflist element is a list of the final RDF files, while the slotlist element is a list of the slots which are written to the RDF files. Slots can be written to multiple RDF files; they're associated with the RDF files by the idxlist attribute, whose value is a comma-separated list of RDF file indices. RiverWare initializes the RDF DMI and mines its data structures to generate rdflist and slotlist.

<rdflist num="2">

<rdf name="R:\\CRSS\\results\\Res.rdf" **idx**="0"/>

<rdf name="R:\\CRSS\\results\\Salt.rdf" idx="1"/>
</rdflist>
<slotlist>
<slot name="Powell.Outflow" idxlist="0,1"/>
<slot name="Powell.Storage" idxlist="0"/>
</slotlist>

The envlist element specifies RiverWare's runtime environment; RIVERWARE_HOME is from the version of RiverWare which starts the distributed concurrent run, all others are from the MRM configuration.

<envlist>
<env>RIVERWARE_HOME_516=C:\\Program Files\\CADSWES\\RiverWare 5.1.6 Patch</env>
<env>CRSS_DIR=R:\\CRSS</env>
</envlist>

The tempdir element is from the MRM configuration and is the intermediate directory where the individual simulations write the partial RDF files.

<tempdir>R:\\CRSS\\temp</tempdir>
</distrib>

RiverWare Configuration: The RiverWare configuration defines a RiverWare batch mode invocation. In a distributed concurrent run, DistribMrmCtlr creates the RiverWare configuration; in other contexts a user could create the RiverWare configuration. This example shows a RiverWare configuration from a distributed configuration; not all elements are valid in other contexts. Some elements are from the MRM configuration, the others are provided by DistribMrmCtlr. Key elements, preceded by brief descriptions, are:

<RW>

The simulation computer, from the distributed configuration's mrm element.

<host addr="lc1.usbr.gov" port="27285"/>

RiverWare's executable, from the distributed configuration's app element.

<app>C:\Program Files\CADSWES\RiverWare 5.1.6 Patch\riverware.exe</app>

Creates the batch script file; the name attribute is provided by DistribMrmCtlr, with each individual simulation having a unique name.

<script name=R:\\CRSS\\temp\\script0.rcl">

Adds the OpenWorkspace command to the batch script file.

<openws>R:\\CRSS\\model\\CRSS.mdl</openws>

Adds the StartController command to the batch script file; firstTrace, numTrace and ctlFile are StartController !MRM options. **HERE (BatchMode.pdf, Section 4.3)**. *The config attribute and firstTrace and numTrace elements are from the distributed configuration. The ctlFile element's value is provided by DistribMrmCtlr, with each individual simulation having a unique name. DistribMrmCtlr creates the control file from the distributed configuration's rdflist and slotlist elements.*

<start>
<mrm config=”Powell Mead 2007 ROD Operations”>
<firstTrace>1**</firstTrace>**
<numTrace>200**</numTrace>**
<ctlFile>R:\\CRSS\\temp\\control0.ctl”**</ctlFile**>
</mrm>
</start>

Adds the CloseWorkspace command to the batch script file.

<close/>
</script>

Specifies RiverWare’s output file; the value is provided by DistribMrmCtlr, with each individual simulation having a unique name. The value is used with the new “--remout” option, described below.

<output>R:\\CRSS\\temp\output0.log**</output>**

RiverWare’s runtime environment, from the distributed configuration.

<envlist>
<env>RIVERWARE_HOME_516=C:\\Program Files\\CADSWES\\RiverWare 5.1.6 Patch**</env>**
<env>CRSS_DIR=R:\\CRSS**</env>**
</envlist>
</RW>

GenApp Configuration: The GenApp defines a generic application. In the standard case, it will executed the perl script that combines the individual RDF files into one RDF file.

```
<GenApp>
  <app>C:\\Perl\\bin\\perl.exe</app>
  <arglist>
    <arg>R:\\CRSS\\bin\\CombineRdf.pl</arg>
    <arg>-o</arg>
    <arg>R:\\CRSS\\results\\Res.rdf</arg>
    <arg>R:\\CRSS\\temp\\Res.*.rdf</arg>
  </arglist>
  <output>%scratchfile%</output>
  <envlist>
    <env>CRSS_DIR=R:\\CRSS</env>
  </envlist>
</GenApp>
```

Configuration Output Options: Both the RW configuration and the GenApp configuration specify "output" files - for RW configurations, it's the script file, the control file and the output file; for GenApp configurations, it's the output file. If the controller computer and the simulation computers have access to the same network directories then the files can be created in a network directory by naming them, e.g.:

<output>R:\\CRSS\\temp\output0.log**</output>**

If they don't have access to the same network directories then the files must be created on the simulation computer, and it's possible the controller computer (where the files are specified) won't know where to create the files on the simulation computer. To accommodate this, "output" files can have the following special values:

- %tempfile% - The file is a temporary file which persists.
- %scratchfile% - The file is a temporary file which is removed.
- %devnull% - No file is created (the equivalent of redirecting a process's output to /dev/null).

A temporary file is a guaranteed unique file in a temporary directory. (The temporary directory might vary, but on Windows it's commonly C:\WINDOWS\Temp.)

6.4.5 Combining RDF Files With CombineRdf.pl

Each simulation writes an intermediate RDF file which is a portion of the final RDF file; when all simulations have finished the intermediate files are combined into the final file. A Perl script, CombineRdf.pl, combines the intermediate RDF files. Its syntax is:

CombineRdf.pl <number of simulations> <number of traces> <intermediate RDF files> <final RDF file>

DistribMrmCtlr invokes the Perl script, so a user needn't know the specifics of it.

Note that the choice of Perl requires that Perl be installed on the controller computer. If this is not desirable, the program could easily be re-implemented in C++.