Oracle Utilities Live Energy Connect

RTI Confirugation Manager User Guide

Release 6.3.4.0.1

September 2020

Oracle Utilities Live Energy Connect RTI Confirugation Manager User Guide, Release 6.03.004

Revision History

Author	Date	Description	Version
Mary Horvath	12/22/2016	Initial draft.	.5
Mary Horvath	3/29/2017	Update format.	.6
Mary Horvath	9/26/2018	Revised the ICCP Tutorial.	.8
Mary Horvath	5/2/2019	Made some minor adjustments.	1.0
Mary Horvath	9/19/2019	Improved the introduction to the ICCP Tutorial.	1.1

CONTENTS

Welcome

LiveData RTI Configuration Manager™ enables you to recreate your real world data network in a LiveData RTI configuration. This configuration specifies how RTI Server is to move and transform data relating to a specific set of points.

There are many different types of points, such as DNP and ICCP. These points represent inputs from and outputs to the outside world. Each **Point**[1] contains a small amount of information. RTI Server moves these points through multiple connected nodes to an RTI Server output. As RTI Server moves a point from node to node, each node can transform the information associated with the point.

The Welcome chapter gives an overview of RTI Configuration Manager, provides definitions of basic terminology, and gives a preview of the next five chapters.

RTI Configuration Manager Mappings

RTI Configuration Manager uses Virtual Manufacturing Devices (VMDs) to map devices from the outside world into LiveData RTI's internal model.

VMDs

A **VMD** is a container of nodes. Each VMD is associated with a specific type of communications protocol or interface. Most VMD types are intended to handle communications in and out of RTI Server using a particular communications protocol or interface, such as ICCP, DNP-3, Modbus, database access, or others. Thus, a VMD can map a device from the outside world to LiveData RTI's internal variable model, which allows RTI Server to capture, transform, and route data to other devices, systems, or applications in a form that the other device, system, or application can understand. Each instance of a VMD has a network address (that can be configured to be inaccessible).

Node Table, Legend, and Prototype Configuration

RTI Configuration Manager creates a virtual view of the resources in your configuration in the **Node Table**. It allows you to see a sampling of the nodes in your configuration so that you can verify that the configuration is operating correctly from a laptop or a desktop computer.

The Node Table represents each resource in your configuration as a VMD, a node, or a connector between the nodes as shown in Figure 1.

[1] 1. A point is an independent datum that passes through an RTI configuration. For example, a single ICCP node might receive a value (datum) from the ICCP peer, then pass the value through multiple nodes (which might perform operations on the datum) in multiple VMDs, and finally output the value to another device, such as a DNP device. Although the point would travel from an input node to various other nodes and exit through an output node, in this definition of a point, this datum would refer to a single point. 2. A point represents a single input or output value, or an intermediate value, that is controlled by RTI Server. A hard point represents an actual input or output within the system, while a soft point results from logic and math operations applied to other points. Many of the points within a configuration are soft points, containing intermediate values. RTI Configuration Manager removes the distinction between hard and soft points.

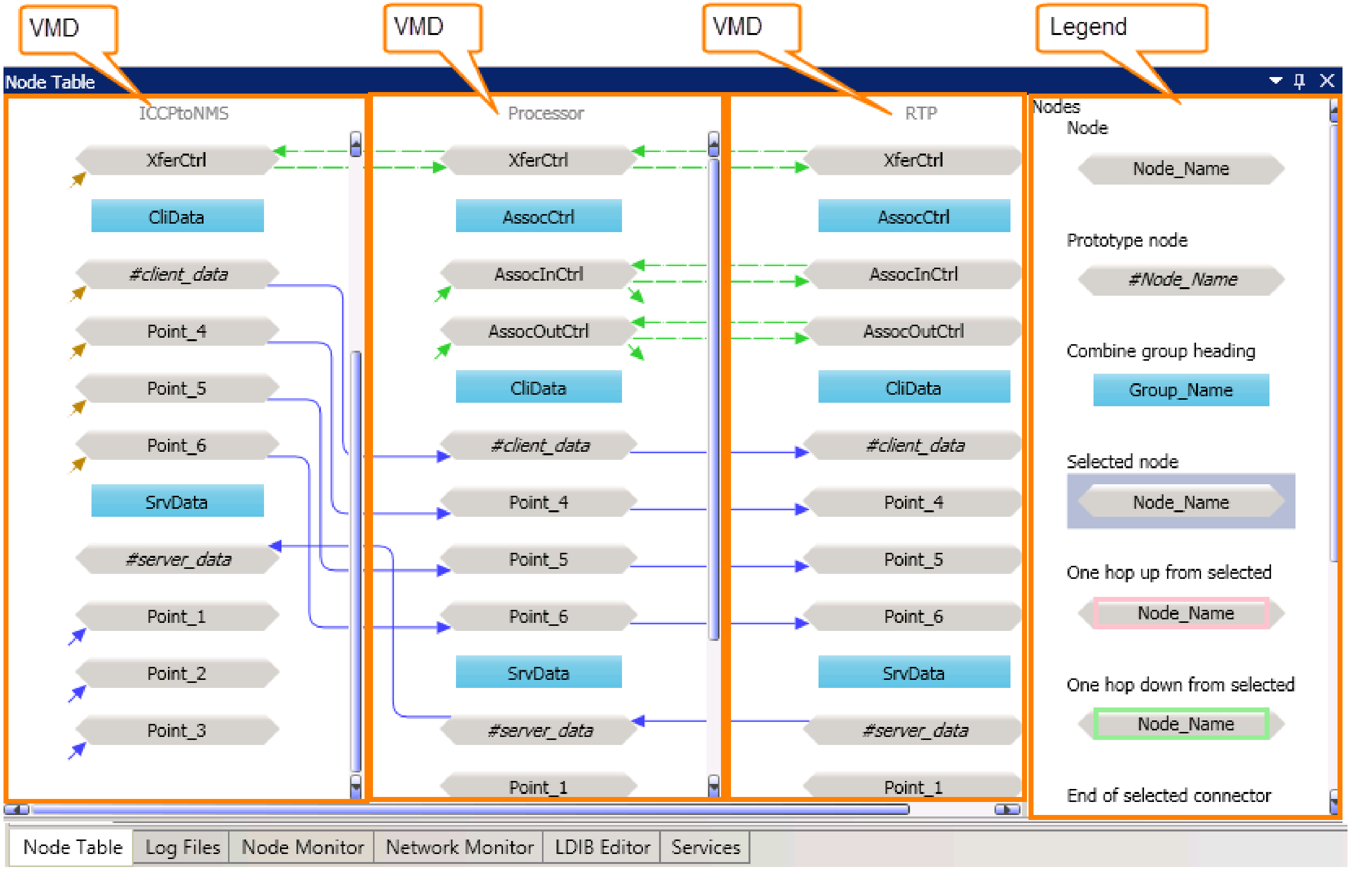


Figure 1: Node Table with its VMDs and Legend Outlined in Orange

Each VMD name in a configuration is placed at the top of a column in the Node Table. The VMDs shown in Figure 1 are ICCPtoNMS, Processor, and RTP. The column below each VMD name shows the VMD's prototype nodes and non-prototype nodes.

Legend

The legend in Figure 1 shows the representation of nodes, prototype nodes, and connectors in the Node Table. The following sections define and describe the interrelationship among the following items in the legend and how they are used to create a prototype configuration:

- Nodes
- Prototype nodes
- Connectors

Nodes

Node Node_Name appears first in the legend. A node is the fundamental processing unit in a configuration; it can interface with the outside world, modify a point's information, or supervise the processing of other nodes.

The node type indicates the role of the node in the configuration.

- A setup node affects the processing, routing, or processing and routing of multiple points. Setup nodes are for internal RTI Server use, so you will not need to do much with them.
- An input node receives data from one point in the physical world via a connection to a remote VMD or script.
- A filter node is an intermediate location in a point's path between two other nodes. A filter node may modify or store, or modify and store data. To modify data, various functions or scripts can be used.
- An output node sends data from one point to the physical network via a connector to a remote VMD or script.

Nodes that store their data as it is processed are monitorable, allowing you to display the data in the node at the current time. Whether or not a node stores its data depends on how it is defined.

Connectors

Each node in your network can be connected to other nodes of various VMDs. A connector is represented by an arrow between two nodes. There are different types of connectors. The Node Table distinguishes between these types by color and by the line that is used (solid, dotted, or dashed). The legend displays each type of these connectors as shown in Figure 2.

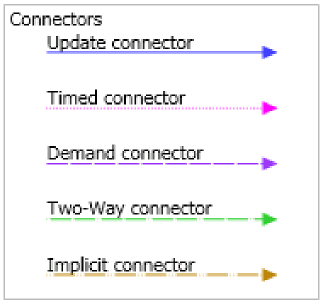


Figure 2: Legend of Connectors

- An **update connector** subscribes to receive notifications from the source node. For storage type nodes, a write to the node will notify the connector. For input type nodes, the notify operation (if supported) notifies the connector. Otherwise, a write operation from the external device notifies the connector.
- A **timed connector** is designed to poll for data periodically by reading from the source node. You can specify the poll rate at the time you create the connector.
- A **demand connector** reads from its source node whenever the destination node reads from the connector. Thus, it perpetuates a read operation. For example, if a time-based connector reads from an Add node, then the Add node will read from each of its inputs; demand-based connectors would be needed at both inputs.
- A **two-way connector** is used in special cases where data moves in both directions. For example, the association control nodes pass association status in one direction, and association control information in the other direction.

Prototype Configuration

Most RTI configurations involve performing similar operations on large numbers of points. It is not practical for an engineer to individually create the nodes and connectors for each of the points. Instead, a LiveData Utilities Professional Services engineer creates a **Prototype Configuration** as a model of the actual configuration. The nodes and connectors for individual points are represented as prototype nodes and prototype connectors.

These prototype nodes are then replicated to form the actual nodes when you load a batch file.

You can create a prototype configuration by using the Templates panel of Configuration Manager; however, a LiveData Utilities Professional Services engineer usually creates the configuration for you and saves it in a .db file, which you then import into the Configuration Manager at your site. See Import a Configuration on page 12 for more information.

Note: In order to create a complete representation of your dataflow, you need to load a batch file to instantiate the prototype nodes and connectors.

Node and Connector Categories

Nodes are categorized according to how, when, and where they are created and used.

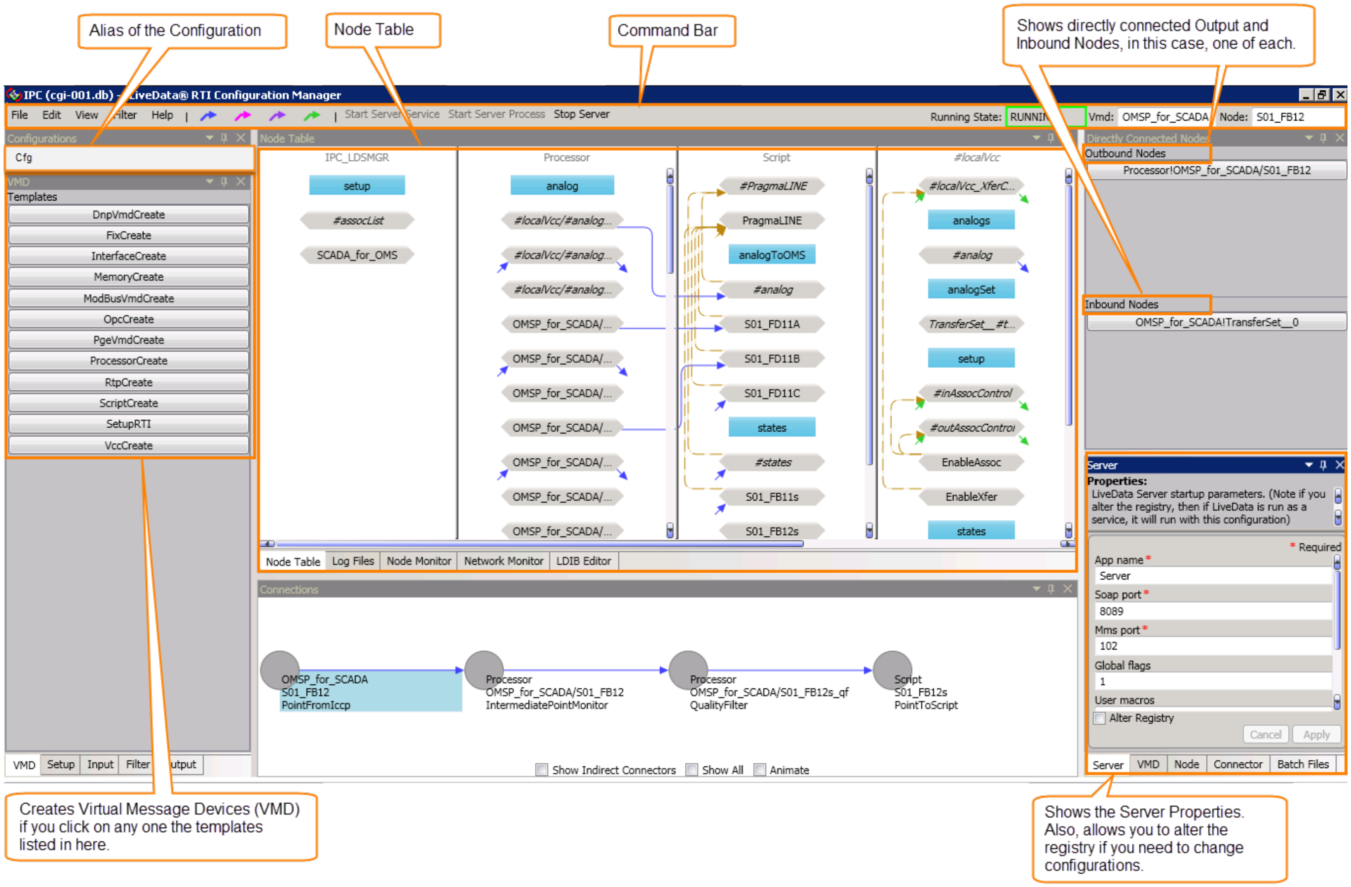
- A **non-batch node** is a permanent part of the configuration, not affected by loading batch files. The parameters of non-batch nodes are passed by Configuration Manager to RTI Server.
- A **prototype node** provides a template from which batch nodes can be created by loading one or more batch files. The name of a prototype node starts with a pound(#) sign and appears in italics. It exists only in Configuration Manager, and not in the configuration executed by RTI Server. When a **batch file** is loaded, each prototype node is replicated many times to create batch nodes of the particular node type.
- A **batch node** is a node created from a prototype node by loading one or more batch files. Configuration Manager passes batch nodes to RTI Server in order to create all the nodes in your configuration.

Batch File

A **batch file** specifies the details of the points in a configuration. A batch file provides the information necessary to replicate prototype nodes and prototype connectors into large numbers of batch nodes and batch connectors when the batch file is loaded into Configuration Manager. There are generally multiple tables in a batch file, where each table has a header row that links to prototype nodes, and table body rows that specify parameters applying to the batch nodes for each point. The batch connectors for each point are specified by the creation of prototype connectors to and from nodes within Configuration Manager. These connectors are not specified within the batch file.

Overview of Configuration Manager

Figure 3 shows the opening view of RTI Configuration Manager and provides a brief description of the default layout of RTI Configuration Manager.



Properties:
LiveData Server startup parameters. (Note if you alter the registry, then if LiveData is run as a service, it will run with this configuration)

* Required

App name *
Server

Soap port *
8089

Mms port *
102

Global flags
1

User macros

□ Alter Registry

Cancel Apply

Alias of the Configuration

Node Table

Command Bar

Shows directly connected Output and Inbound Nodes, in this case, one of each.

Creates Virtual Message Devices (VMD) if you click on any one the templates listed in here.

Shows the Server Properties. Also, allows you to alter the registry if you need to change configurations.

Figure 3: Overview of RTI Configuration

Aliases

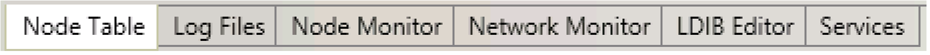
RTI Configuration Manager utilizes aliases in which you can import your prototype configuration, or build configurations within RTI Configuration Manager. There are three aliases that come with RTI Configuration Manager:

- **cfg** is the **Alias**[1] in which you can import a configuration.
- **test1** is for testbed configurations that are intended to simulate other devices and interact with the configuration that is running under the "cfg" alias.
- **test2** is also for testbed configurations.

You can select an alias from a drop-down list in the top left corner of RTI Configuration Manager or create a new alias. In this example, the alias is "Cfg" as shown in Figure 3. See Alias Selection and Creation on page 53 for more information.

Tabs under the Node Table

You can change the view of each section by selecting a different tab. For example, instead of seeing the Node Table in the middle of the screen, you can click on any of the tabs displayed below it as shown below.



[1] An alias is an internal configuration buffer. A configuration is imported from a file into this configuration buffer, named "cfg". While a configuration is in this buffer, RTI Configuration Manager can access it, allowing you to edit the configuration and execute the configuration on RTI Server. For testing purposes, there is a need for more than one active configuration so that you can test various improvements before incorporating them into the configuration. These testing areas are also in internal configuration buffers, usually called "test1" and "test2".

Each tab will provide different information about your configuration:

- Log Files will show you a running log of an active configuration.
- Node Monitor allows you to monitor the information that passes through a node or nodes.
- Network Monitor shows MMS association statuses for each VMD in your configuration.
- LDIB Editor shows an editor where you can to see if your ICCP parameters are correct and change them if needed.
- Services shows each alias that has a service associated with it. From here, you can start or stop each service.

Preview

The subsequent chapters discuss how to:

- Import your LiveData configuration
- Load a batch file
- Use the Configuration Manager interface
- Go live and monitor your network
- Create a simple configuration

Import a Configuration

The first time that you use RTI Configuration Manager you must import the prototype configuration. LiveData Utilities Professional Services creates the prototype configuration for you, and saves it in database file(.db).

Once the configuration has been imported into an alias, it becomes the active configuration for that alias. It can then be loaded, updated, and saved.

> **Note:** It is recommended that you keep a snapshot of a configuration that is in a good state by saving it in a different file. The active configuration can then be reverted to this state by reloading it in case you encounter a problem.

1. Click on **Import configuration** from the File menu as shown in Figure 4.

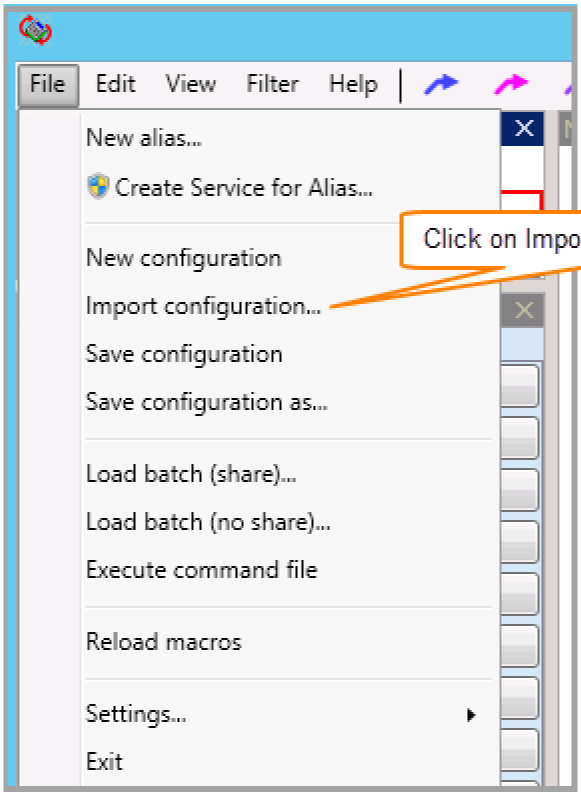
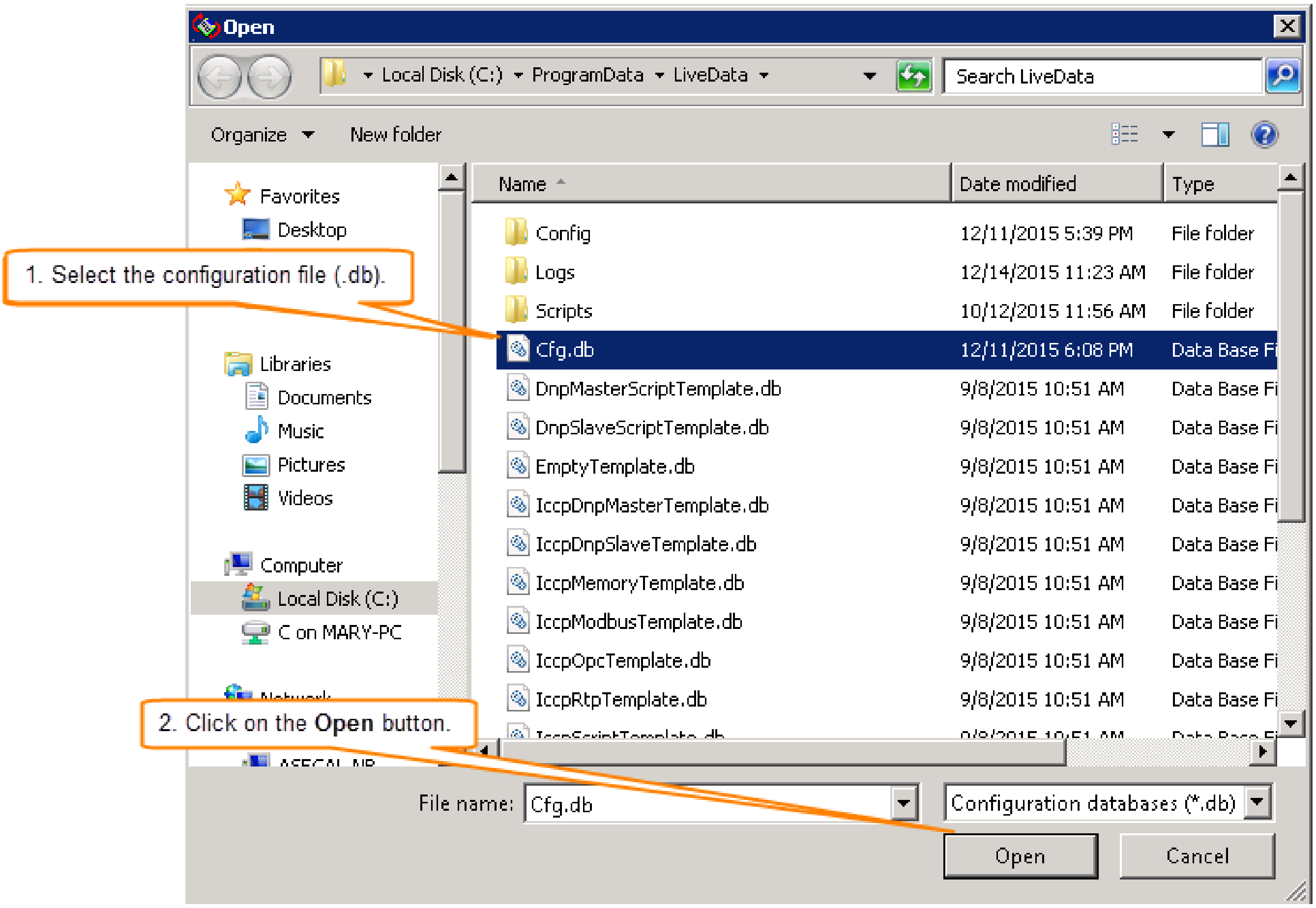


Figure 4: Import the Configuration from the File Menu

Clicking on **Import configuration** brings up a Windows Explorer Window as shown in Figure 5.



1. Select the configuration file (.db).

2. Click on the **Open** button.

Name	Date modified	Type
Config	12/11/2015 5:39 PM	File folder
Logs	12/14/2015 11:23 AM	File folder
Scripts	10/12/2015 11:56 AM	File folder
Cfg.db	12/11/2015 6:08 PM	Data Base Fi
DnpMasterScriptTemplate.db	9/8/2015 10:51 AM	Data Base Fi
DnpSlaveScriptTemplate.db	9/8/2015 10:51 AM	Data Base Fi
EmptyTemplate.db	9/8/2015 10:51 AM	Data Base Fi
IccpDnpMasterTemplate.db	9/8/2015 10:51 AM	Data Base Fi
IccpDnpSlaveTemplate.db	9/8/2015 10:51 AM	Data Base Fi
IccpMemoryTemplate.db	9/8/2015 10:51 AM	Data Base Fi
IccpModbusTemplate.db	9/8/2015 10:51 AM	Data Base Fi
IccpOpcTemplate.db	9/8/2015 10:51 AM	Data Base Fi
IccpRtpTemplate.db	9/8/2015 10:51 AM	Data Base Fi
IccpScriptTemplate.db	9/8/2015 10:51 AM	Data Base Fi

File name: Cfg.db

Configuration databases (*.db)

Figure 5: Select and Open the Configuration File

2. Browse to the configuration that you would like to select, then click on the Open button as shown in Figure 5.

Load a Batch File

The RTI Configuration Manager can load one or more batch files into a prototype configuration to create a complete representation of your dataflow. This representation contains all information needed to run the LiveData RTI Server for a specific data network. Batch files are .csv files that contain tables specifying lists of data point names and parameters, and sometimes lists of virtual device names and parameters.

LiveData Utilities Professional Services will help you to create the batch file. For detailed information about batch files, see LiveData's *Tutorial on Batch Configuration*.

To load a batch file:

1. Select **Load batch (share)** or **Load batch (no share)** from the file drop-down list as highlighted in Figure 6.

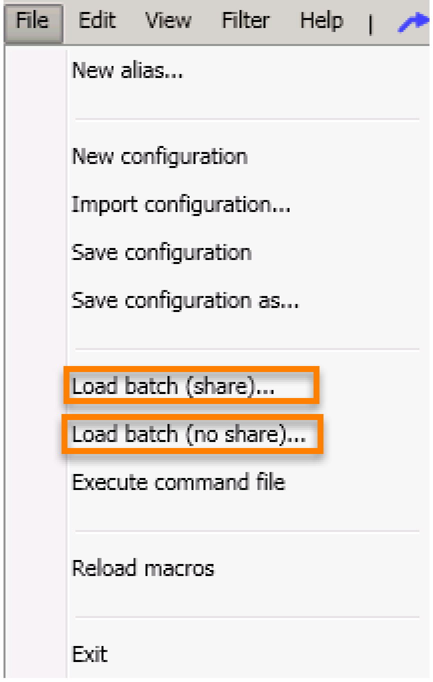
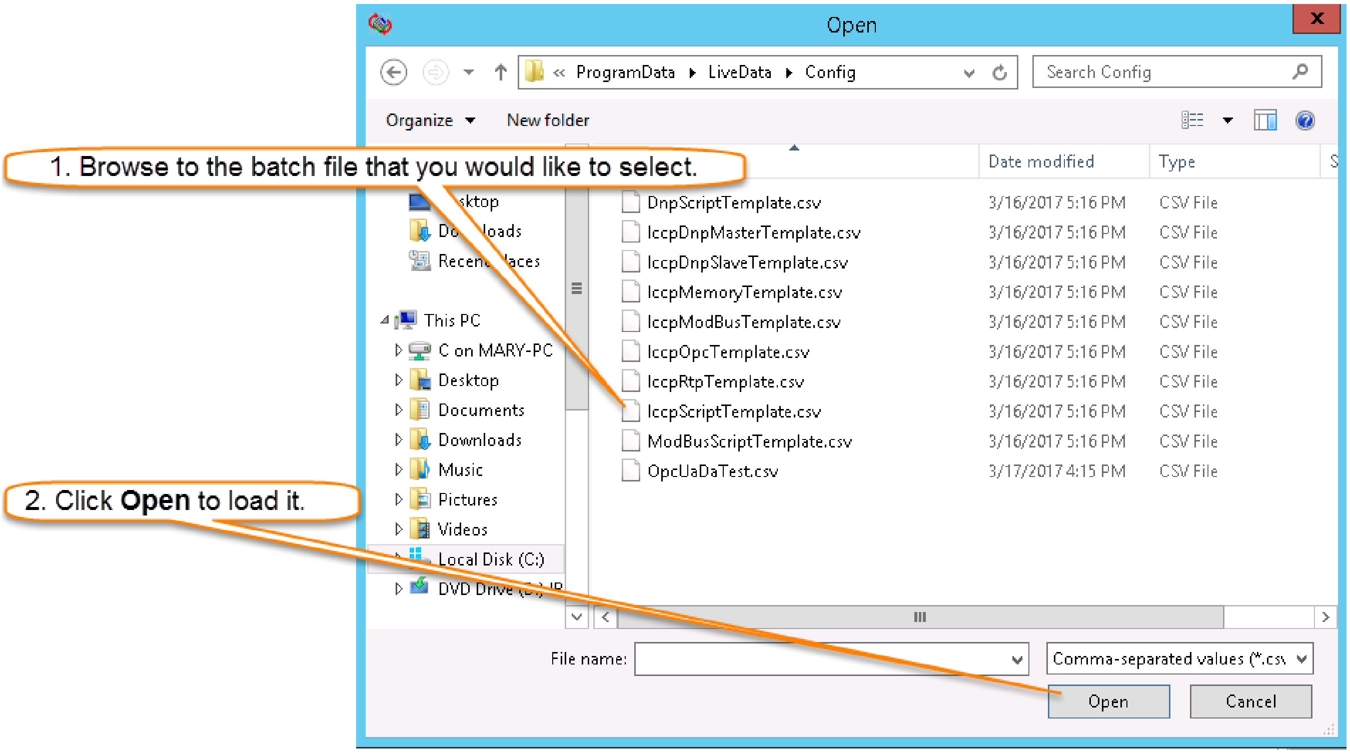


Figure 6: Load a Batch File from the File Menu

Load batch (share) allows two or more batch files to share one or more prototype nodes.

Load batch does not allow any of the prototype nodes declared in the batch file to be shared.

Clicking either of the **Load batch** commands brings up Windows Explorer.



1. Browse to the batch file that you would like to select.

2. Click **Open** to load it.

Name	Date modified	Type
DnpScriptTemplate.csv	3/16/2017 5:16 PM	CSV File
IccpDnpMasterTemplate.csv	3/16/2017 5:16 PM	CSV File
IccpDnpSlaveTemplate.csv	3/16/2017 5:16 PM	CSV File
IccpMemoryTemplate.csv	3/16/2017 5:16 PM	CSV File
IccpModBusTemplate.csv	3/16/2017 5:16 PM	CSV File
IccpOpcTemplate.csv	3/16/2017 5:16 PM	CSV File
IccpRtpTemplate.csv	3/16/2017 5:16 PM	CSV File
IccpScriptTemplate.csv	3/16/2017 5:16 PM	CSV File
ModBusScriptTemplate.csv	3/16/2017 5:16 PM	CSV File
OpcUaDaTest.csv	3/17/2017 4:15 PM	CSV File

File name:

Comma-separated values (*.csv)

.

Figure 7: Select and Open the Batch File

2. Browse to the batch file that you would like to select, then click the **Open** button as shown in Figure 7.

For more information about loading batch files, see Batch Files on page 20.

Advanced users who would like an in-depth understanding of batch files may refer to LiveData’s *Tutorial on Batch Configuration*.

Start and Stop Your Configuration

After you have loaded the batch files needed to create all the nodes in your configuration, you are ready to start the flow of data from the physical data network.

In order to start the data flow into the model described in your configuration, you must start RTI Server Service.

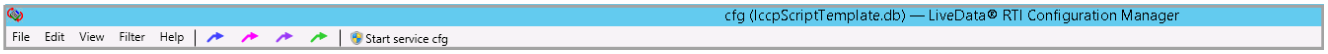
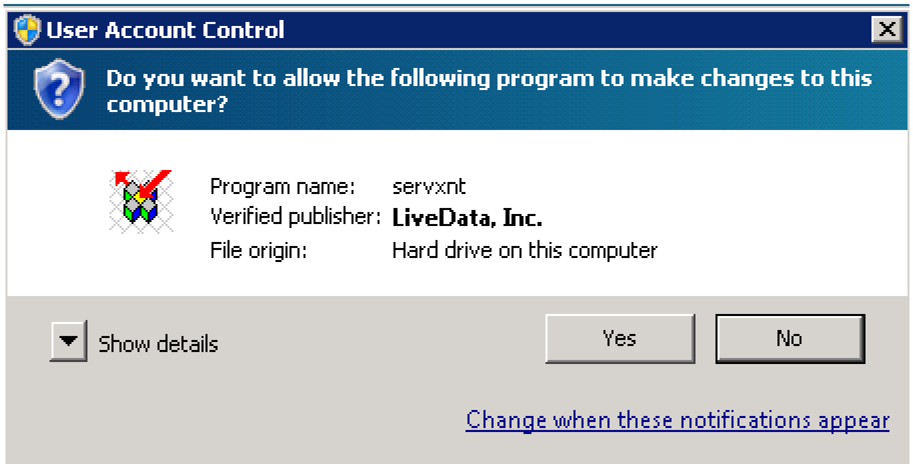

cfg (IccpScriptTemplate.db) — LiveData® RTI Configuration Manager
File
Edit
View
Filter
Help
Start service cfg

Figure 8: RTI Configuration Manager Command Bar

To Start RTI Server as a Windows Service:

1. Click the **Start Service** command on the Command Bar at the top of the RTI Configuration Manager window as shown in Figure 8. The beginning of the command is marked by a security shield, and the name of the selected alias (cfg in this example) marks the end of the command as shown below:

If you are not running RTI Configuration Manager with Administrative privileges, a pop-up message is displayed, asking if you want to allow the servxnt program (RTI Server) to make changes to your computer as shown in Figure 9. If you are running with Administrative privileges, RTI Server will start without first issuing the pop-up message.



Do you want to allow the following program to make changes to this computer?

Program name: servxnt

Verified publisher: **LiveData, Inc.**

File origin: Hard drive on this computer

Figure 9: User Account Control Message

2. Click **Yes** to start RTI Server if you see the pop-up.

To stop running RTI Server:

1. Click the **Stop Service** command on the command bar at the top of the RTI Configuration Manager window. The beginning of the command is marked by a security shield, and the name of the selected alias (cfg in this example) marks the end of the command as shown below:

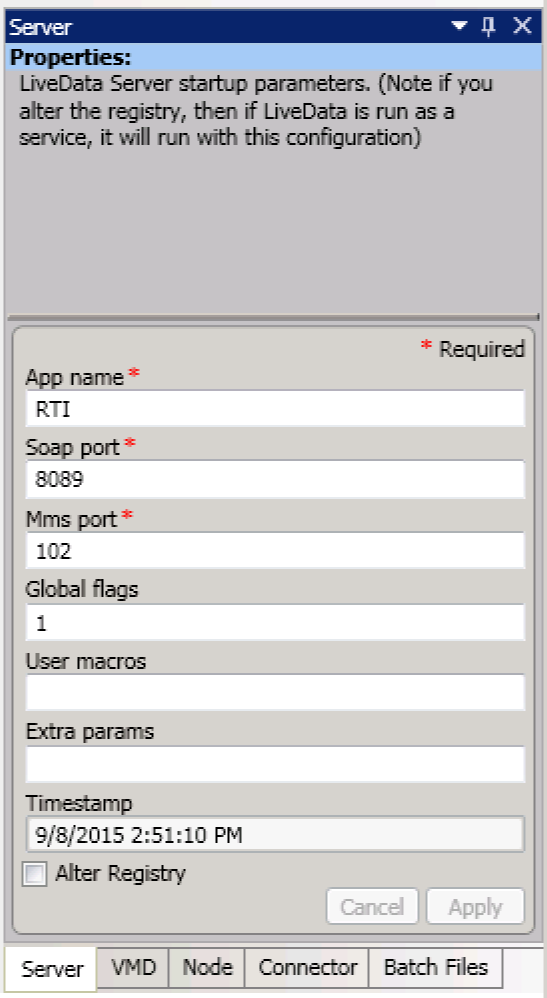
Learn More about the Interface

This section describes the Properties panel, Command Bar, and each of the panels within RTI Configuration Manager:

- Node Table
- Log Files
- Node Monitor
- Network Monitor
- LDIB Editor
- Services
- Templates panel
- Configuration panel

Properties Panel

Figure 10 shows the Properties panel that is located in the lower right corner of RTI Configuration Manager. Here you can view and modify the property values of RTI Server, VMDs, nodes, connectors, and batch files. Figure 10 displays the first Properties panel tab "Server," which contains the startup parameters of RTI Server.



Properties:
LiveData Server startup parameters. (Note if you alter the registry, then if LiveData is run as a service, it will run with this configuration)

* Required

App name *: RTI

Soap port *: 8089

Mms port *: 102

Global flags: 1

User macros:

Extra params:

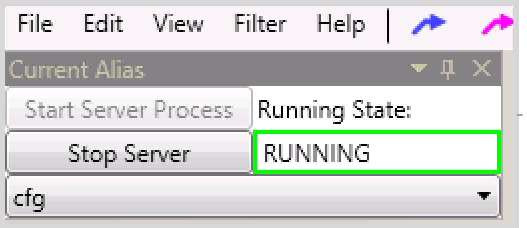
Timestamp: 9/8/2015 2:51:10 PM

□ Alter Registry

Figure 10: Properties Panel

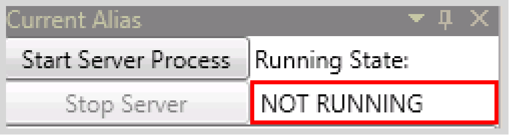
To change the startup parameters of RTI Server:

- Stop RTI Server by clicking on the **Stop Server** command from the Command Bar at the top of RTI Configuration Manager or from the Current Alias panel.



Running State: RUNNING

- Wait until the Running State indicates that RTI Server is no longer running.



Running State:
NOT RUNNING

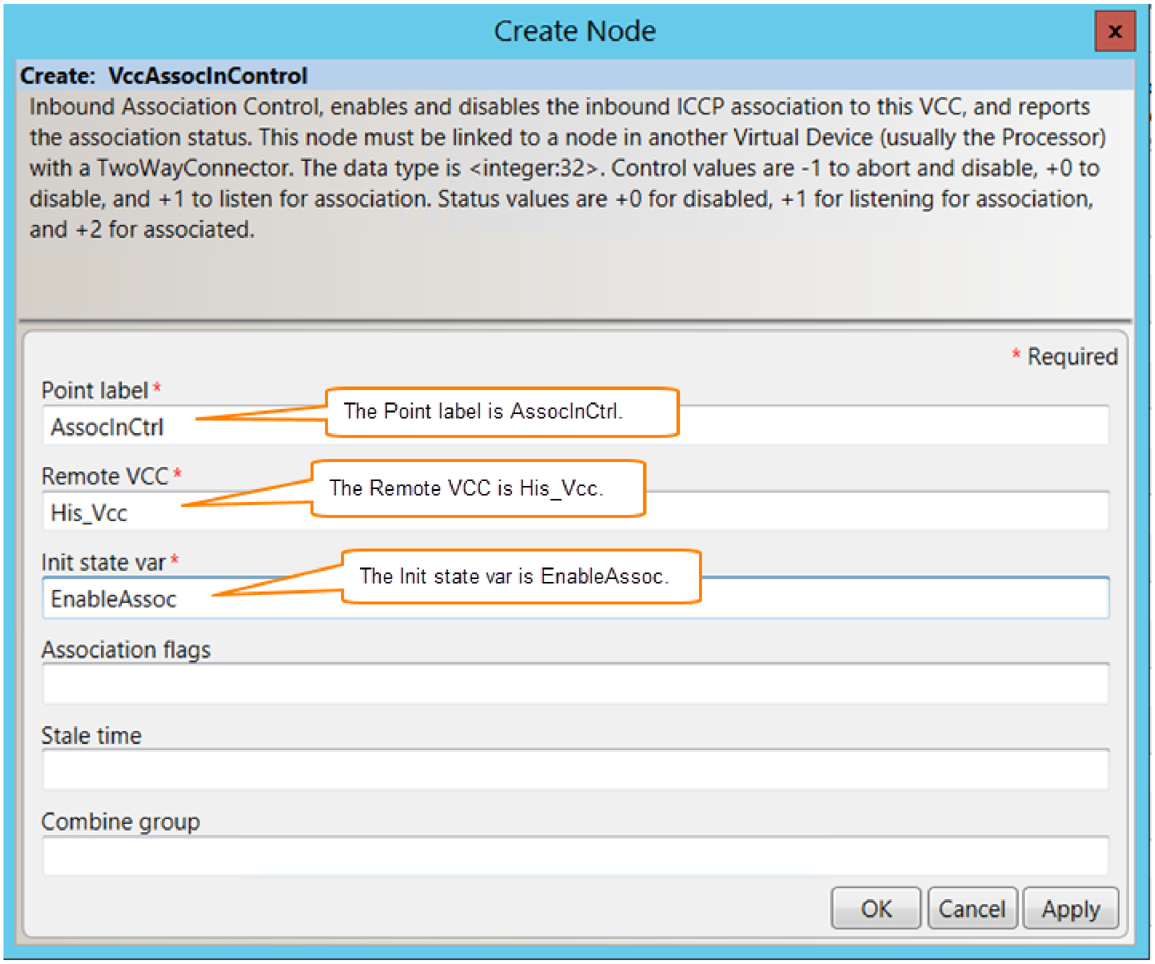
- Make the necessary changes in the fields displayed from the Server tab as shown in "Properties Panel" on the previous page
- Check the Alter Registry checkbox.
- Restart RTI Server by clicking on **Start service cfg** from the Command Bar. "cfg" is the name of the alias in this example. If you were using the alias "test1," the **Start service** button would end with test1, as in **Start service test1**.

To view the properties of a VMD, prototype node, node, group heading, or connector in the Node Table:

- Click on the item of interest in the Node Table, such as the node AssocInControl.

  The Properties panel will show you a definition of the prototype node, node, group heading, or connector that you have selected and provide a form that contains its properties.

Figure 11 shows an example in which the selected node is AssocInControl.



Create: VccAssocInControl

Inbound Association Control, enables and disables the inbound ICCP association to this VCC, and reports the association status. This node must be linked to a node in another Virtual Device (usually the Processor) with a TwoWayConnector. The data type is <integer:32>. Control values are -1 to abort and disable, +0 to disable, and +1 to listen for association. Status values are +0 for disabled, +1 for listening for association, and +2 for associated.

* Required

Point label *: AssocInCtrl

Remote VCC *: His_Vcc

Init state var *: EnableAssoc

Association flags

Stale time

Combine group

The Point label is AssocInCtrl.
The Remote VCC is His_Vcc.
The Init state var is EnableAssoc.
OK
Cancel
Apply

Figure 11: Properties Panel for the Node AssocInControl

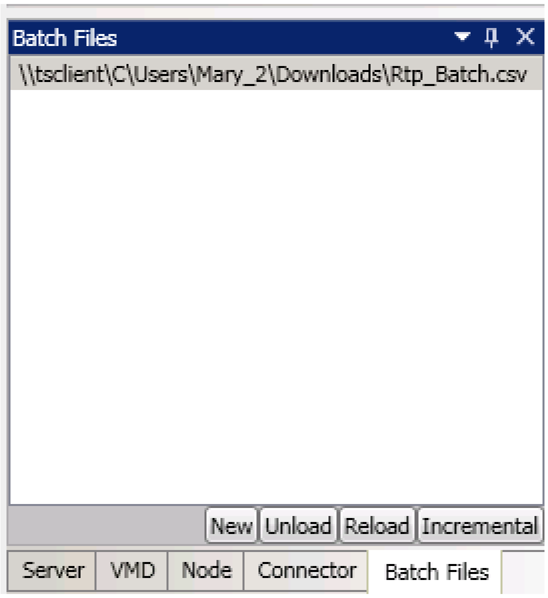
You can change the values of any of the fields under Point Label, such as the Association Flags, Combine Group, InitStatVar, and Remote VCC, if necessary, then press the **Apply** button.

You can make changes in any of the fields of the other tabs as well.

Note: If you make changes to a prototype node, connector, or in anyway change the layout of the configuration shown in the Node Table, you need to stop the server and then reload the batch file. Read the next section on batch files for more information.

Batch Files

Whenever you or a LiveData Utilities Professional Services engineer makes changes to a batch file to add or delete a node or nodes, or if you change any of the nodes or connectors in the Node Table, you will need to reload the batch file.



\\tsclient\C\Users\Mary_2\Downloads\Rtp_Batch.csv

Figure 12: Batch Files Panel

To reload a modified batch file

Reloading brings in the existing batch file if you have modified it since you last loaded it.

- Click on the Reload button to bring in the revised batch file.

Note: You do not need to stop the service in order to reload a batch file. However, if you are loading more than one batch file, it is recommended that you stop the service before reloading them.

To create a new batch file

- Click on the **New** button to create a new batch file. This will open the LiveData\Config directory in Windows Explorer. RTI Configuration Manager, by default, names the file New Batch.csv.
- Press the **Open** button to edit it in Notepad.
- Save the file with **File>Save As** to give the file an appropriate name.
- Stop RTI Server by clicking on the **Stop server** button from the Command Bar.
- After the server has stopped in Windows Task Manager, select the **Load Share** or **Load No Share** depending on how you would like to load the batch file.
- Restart RTI Server by clicking on the **Start service** button from the Command Bar.

To load only the parts of the batch file that have changed

The following procedure saves time if you have a particularly large batch file.

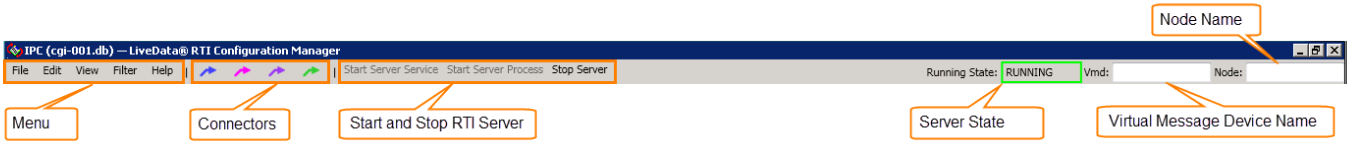
- Stop RTI Server by clicking on the **Stop service** button from the Command Bar.
- Click on the **Incremental** button.
- Restart RTI Server by clicking on the **Start service** button from the Command Bar.

To unload a batch file

- Select the batch file from the Batch Files panel.
- Click on the **Unload** button to unload it.

Command Bar

The Command Bar is divided into several sections as shown in Figure 13.



Running State: RUNNING

Vmd:

Node:

Figure 13: Sections in the Command Bar

Menu

The menu on the Command Bar provides the following items:

- File
- Edit
- View
- Filter
- Help

File

Figure 14 shows the File menu.

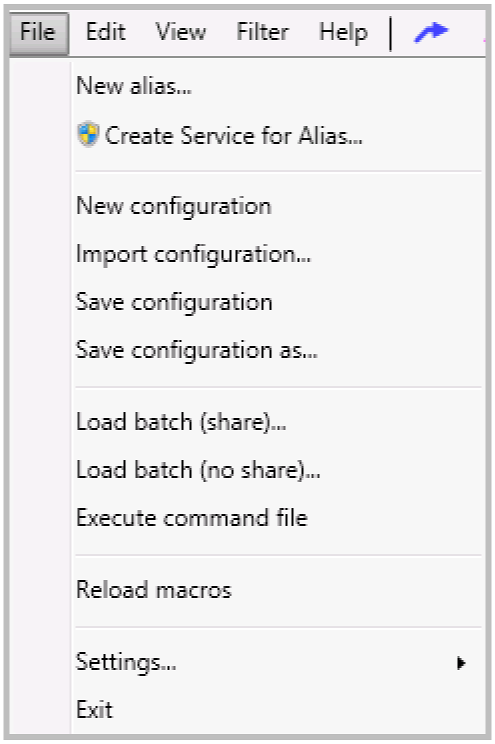


Figure 14: File Menu

From here, you can do any of the following tasks:

- Create a new alias for an existing or new configuration.
- Create a service for a selected alias. (For more information on starting and stopping multiple services, see the section Services on page 49.)
- Create a new configuration. (Consult LiveData Utilities Professional Services.)
- Import a configuration. (See Import a Configuration on page 12 for more information.)
- Save (overwrite) the imported configuration database with its current name. (Not recommended)
- Save a configuration with a new name. (Recommended)
- Load batch files. (See Load a Batch File on page 14 for more information.)
- Execute a command file.
- Reload macros whenever you upgrade to a new version of RTI Configuration Manager.
- Exit from RTI Configuration Manager.

Save and Save configuration as...

You have two options for saving a configuration database file that you have created or imported. If you have imported a configuration database, it is recommended that you use the **Save configuration as...** option because it allows you to save your changes to a new database configuration file while preserving the original configuration database file that you imported. In this way, you have the original configuration database file as a starting point for other configurations.

If you use **Save configuration as...**, you generally rename the file with a descriptive name; however, if you neglect to rename the file, RTI Configuration Manager will rename the file by appending a version number after the original file name, such as **IccpScriptTemplate-001.db**. If you saved the same configuration database file again without giving it a name, using **Save configuration as...**,RTI Configuration Manager would rename the file **IccpScriptTemplate-002.db**.

Note: The alias name has nothing to do with the configuration database file name. It represents a sandbox in which you do your work.

Edit

Figure 15 shows the Edit menu.

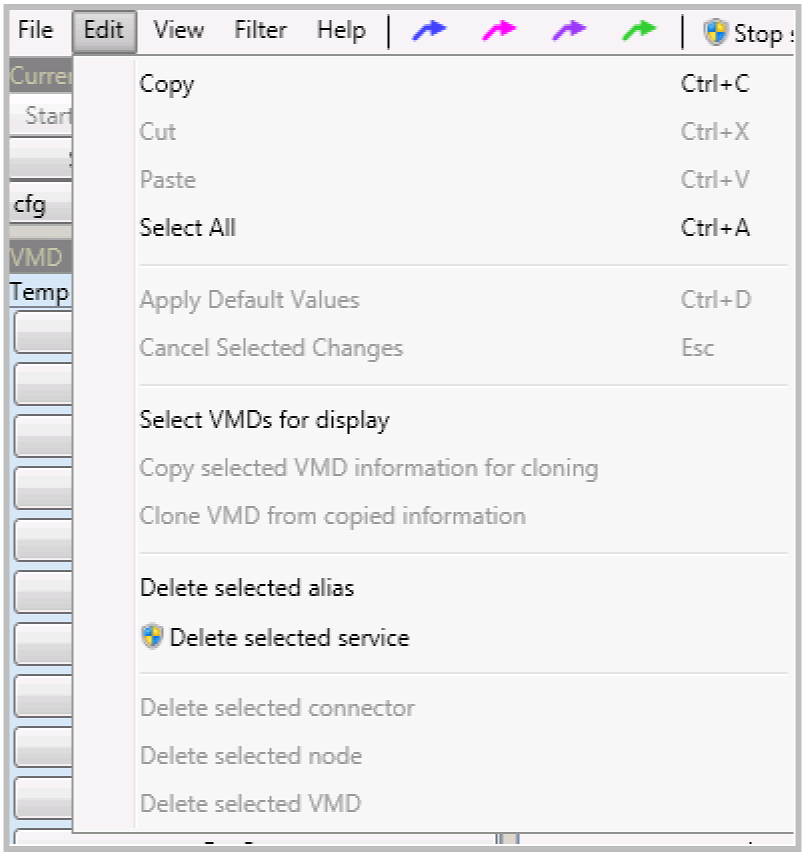


Figure 15: Edit Menu

The Edit menu is useful when you need to change or delete a value in the LDIB Editor or in the Properties panel in the lower right corner of Configuration Monitor. From this menu, you can also delete the selected alias or the selected service.

For more information on the LDIB Editor, see LDIB Editor on page 48.

For more information on the Properties editor, see Properties Panel on page 18.

View

Figure 16 shows the View menu.

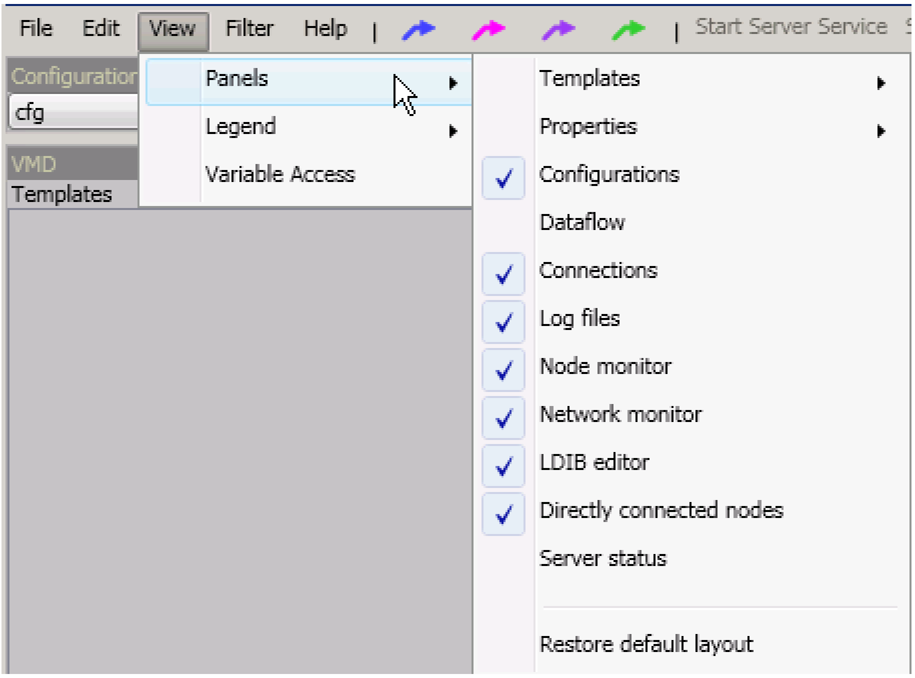


Figure 16: View Menu

Each section within RTI Configuration Manager is a panel. The Node Table, the Properties panel, the LDIB Editor are all panels.

From the panels menu under View, you can change which panels, templates, and properties are to be displayed and determine RTI Server status.

- Select which panels are to be displayed in the foreground of the main panel.
- Select which Templates are to be displayed in the front panel. See Templates on page 26 for more information.
- Select which Properties are to be displayed in the front panel. See Properties on page 27 for more information.
- Select Server Status to display internal diagnostics.
- Restore the default layout.

Templates

Figure 17 shows the Templates menu that is within the panels menu.

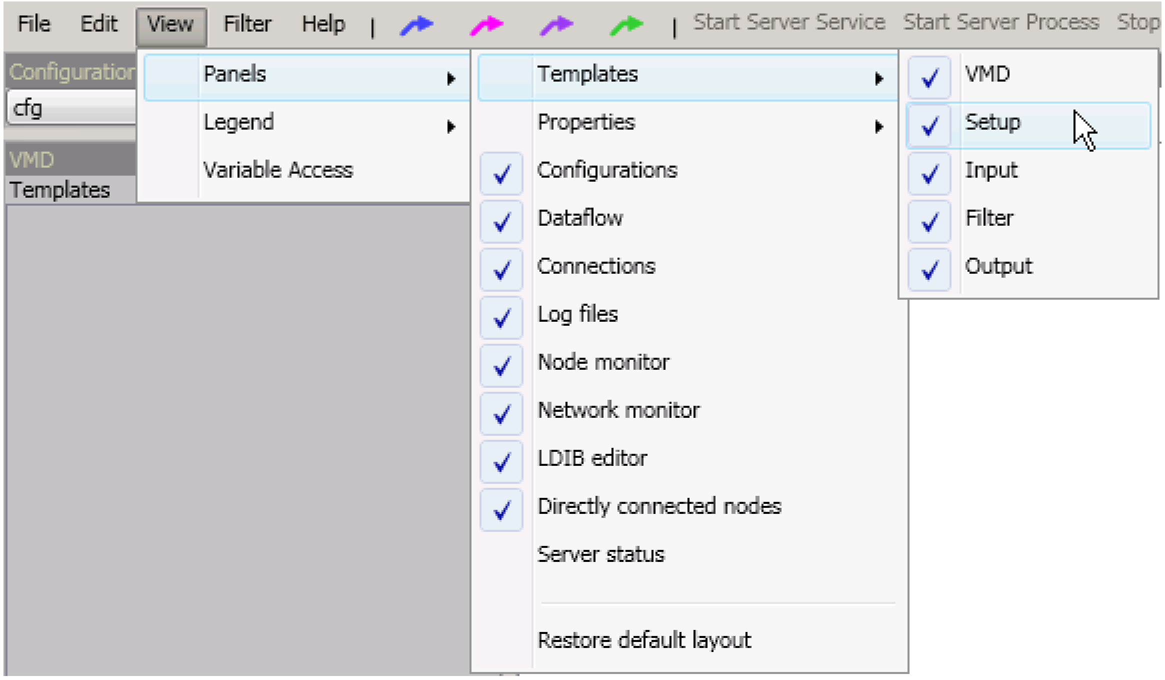


Figure 17: View>panels>Templates Menu

From the Templates Menu, you can select those panels that you need to see or deselect panels that you do not need to see. Click on the checkbox next to a panel to deselect or select it. If you decide that you would like to return to the default layout, select **Restore default layout** at the end of the menu.

Properties

Figure 18 shows the Properties menu that is within the Panels menu.

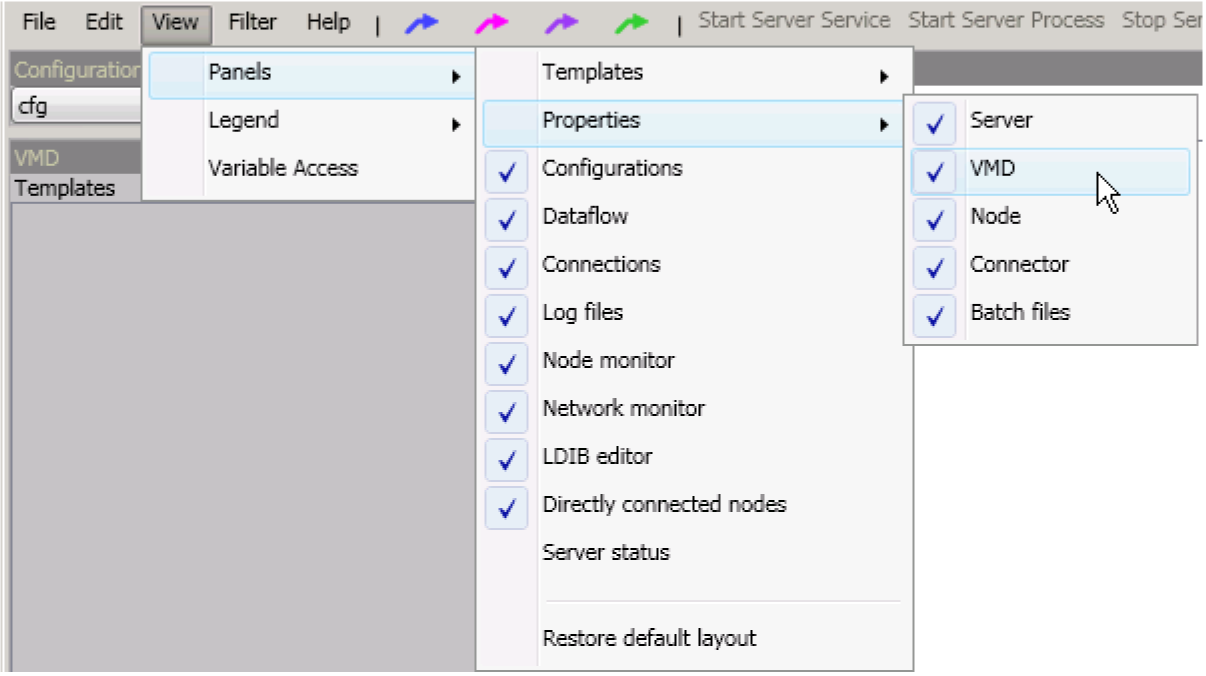


Figure 18: View>panels>Properties Menu

Figure 19 shows the Legend menu of the View menu.

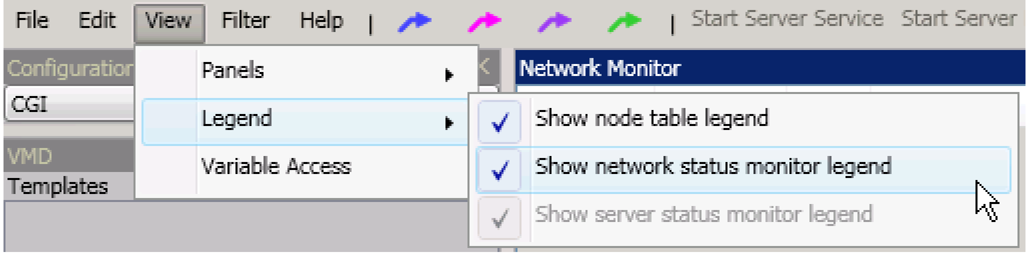


Figure 19: View>Legend Menu

From here, you can

- Show the Node Table legend on the Node Table panel.
- Show the Network Status Monitor legend on the Network Monitor panel.
- Show the service monitor status legend (see Server Status panel on page 57.)

Variable Access

You can monitor variables in the Variable Access panel as shown in Figure 20.

Some nodes can be monitored, which involves RTI Configuration Manager polling a variable that is implemented by that node. You can view monitorable nodes and associated variables with the Variable Access tool.

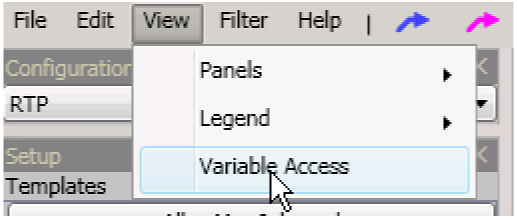
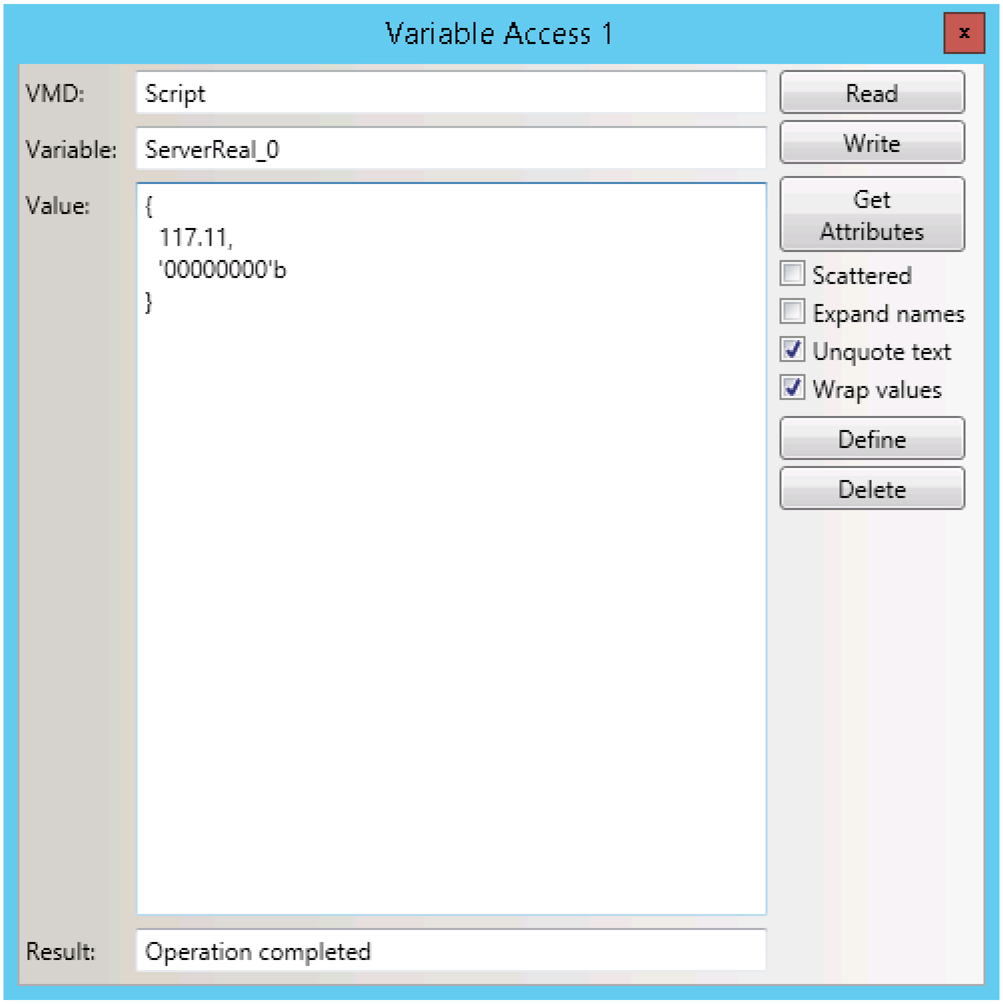


Figure 20: View Menu provides Variable Access

 shows how you can read the value of a particular node by specifying the VMD and the node name, also known as a point. In this case, the VMD is Processor; the Node, which is also part of the internal variable model, is AssocOutCtrl, and its value is +1.

Variables are an integral part of how a configuration is constructed. Technically, a node represents a macro invocation that might create one or more variables. Some nodes do not create any variables, such as some of the Setup nodes. If a node has no variables, you cannot monitor it. Other nodes have exactly one variable, such as IntermediatePointMonitor, which you can monitor. In fact, if you would like to monitor a node that is not monitorable, connect it to an IntermediatePointMonitor node so that you can monitor it indirectly. Other nodes have more than one variable, in which case, Variable Access will display the values of all associated variables.



VMD: Script

Variable: ServerReal_0

Value:

```
{
  117.11,
  '00000000'b
}
```

☐ Scattered

☐ Expand names

☑ Unquote text

☑ Wrap values

Result: Operation completed

Figure 21: Variable Access allows you to Read, Write, Delete, and Get the Attributes of a Node

From here, you can perform any of the following tasks:

- Write to a node.
- Read from a node.
- See a node's attributes.
- Delete a node.
- Check the **Unquote text** checkbox to remove the quotation marks around a string value.
- Check the **Scattered** checkbox to view **scattered variables**[1].
- Check the **Wrap** checkbox so that a structure is formatted for readability, not exceeding the width of the text box as shown in Figure 21.
- Define the attributes of a node.

To write to a node

- Specify the VMD of the node in the VMD field.
- Specify the node name in the Variable field.
- Click the **Read** button to see the current value or values.

[1] Variables that gather data points, which logically should be grouped together, but whose addresses are not in sequence. Once you define a scattered variable, LiveData treats it like a standard structure. Scattered variables can gather different data types from scattered locations in device memory, eliminating the need for control programs to copy and re-cast data. Scattered variables are designated as "vscatter" instead of "variable."

- Change the value or values that need to be changed in the Value field.

  For example, there might be an array of Boolean values, such as [true, true, false, false, false, false, false, false, false, false, false, false, false, false], and you might need to change one of the true values to false. You would make the change in the Values field.
- Click on the **Write** button.

To read from a node

- Specify the VMD of the node in the VMD field.
- Specify the node name in the Variable field.
- Click on the **Read** button.
- Read the node's value in the Value field.

To see a node's attributes

- Specify the VMD of the node in the VMD field.
- Specify the node name in the Variable field.
- Click on the **Get Attributes** button.
- Read the node's attributes in the Value field.

To delete a node

- Specify the VMD of the node in the VMD field.
- Specify the node name in the Variable field.
- Click on the **Delete** button.

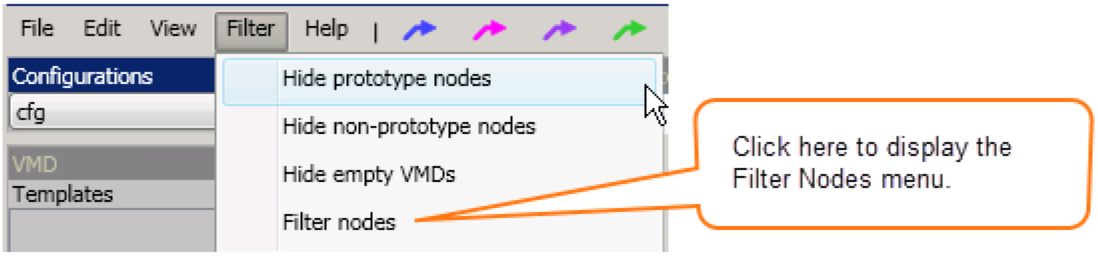
To define the attributes of a node

- Specify the VMD of the node in the VMD field.
- Specify the node name in the Variable field.
- Specify the data type and possibly its initial value.
- Click on the **Define** button.

Note: Instead of defining attributes, most users write to an existing attribute or contact LiveData Utilities Professional Services for help.

Filter

Figure 22 shows the Filter menu.



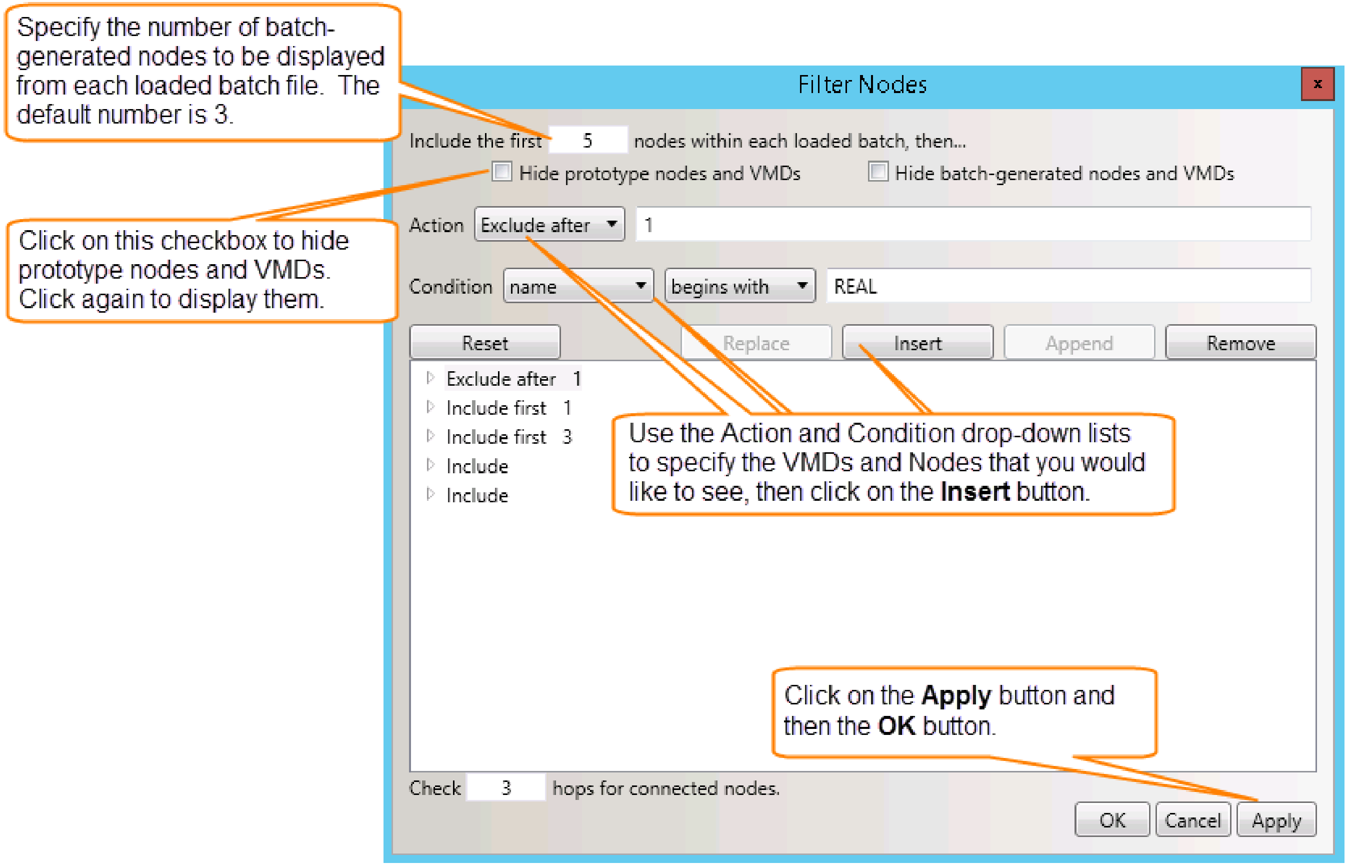
Click here to display the Filter Nodes menu.

Figure 22: Filter Menu

The Filter menu allows you to control which nodes are displayed and which are not in the Node Table. Specifically, you can hide or display any of the following types of nodes and VMDs:

- Prototype nodes and VMDs
- Batch-generated nodes and VMDs

You can also limit the number of batch nodes that are displayed, and pinpoint specific VMDs, prototype VMDs, nodes, batch-generated nodes, and prototype nodes by using the form that is displayed when you choose **Filter nodes**.



Include the first 5 nodes within each loaded batch, then...

□ Hide prototype nodes and VMDs □ Hide batch-generated nodes and VMDs

Action Exclude after 1

Condition name begins with REAL

- Exclude after 1
- Include first 1
- Include first 3
- Include
- Include

Check 3 hops for connected nodes.

Specify the number of batch-generated nodes to be displayed from each loaded batch file. The default number is 3.

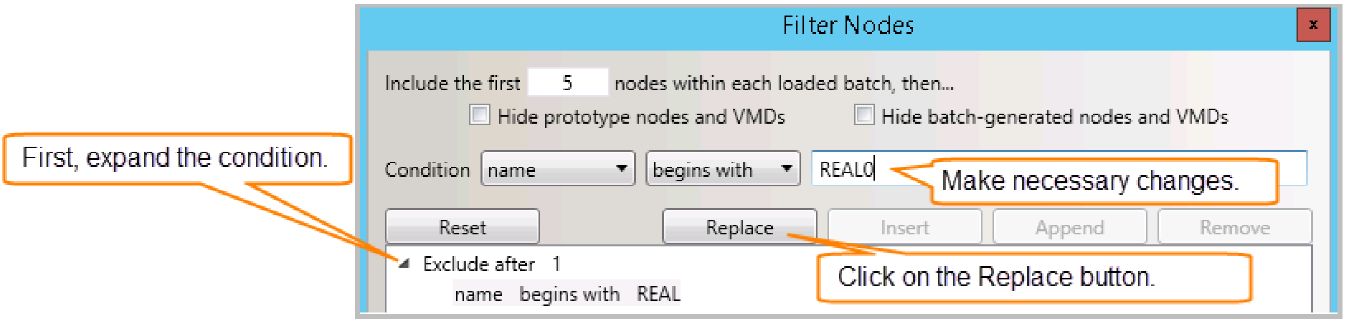
Click on this checkbox to hide prototype nodes and VMDs. Click again to display them.

Use the Action and Condition drop-down lists to specify the VMDs and Nodes that you would like to see, then click on the **Insert** button.

Click on the **Apply** button and then the **OK** button.

Figure 23: Specify the Nodes and VMDs to be Displayed in the Node Table

In addition to the functions pointed out in Figure 23, you can also hide or display batch-generated nodes and VMDs by clicking on the **Hide batch-generated nodes and VMDs** checkbox. You can remove one of the filter conditions by clicking on the condition in the list below the buttons, and then click on the **Remove** button. For example, you could click on "Exclude after 1," and then click on the **Remove** button to remove this condition. Similarly, you can choose a condition, expand the condition by clicking on the pointer to the left of the condition, make any changes to it that are needed, and then click on the **Replace** button as shown in Figure 24.



Filter Nodes

Include the first 5 nodes within each loaded batch, then...

□ Hide prototype nodes and VMDs □ Hide batch-generated nodes and VMDs

Condition name begins with REAL0

Exclude after 1

name begins with REAL

First, expand the condition.

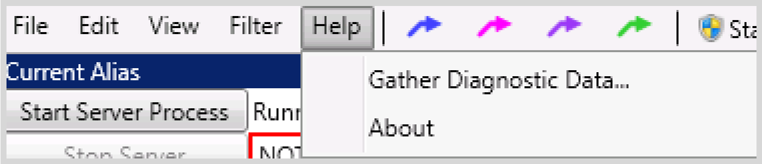
Make necessary changes.

Click on the Replace button.

Figure 24: Edit an Existing Condition

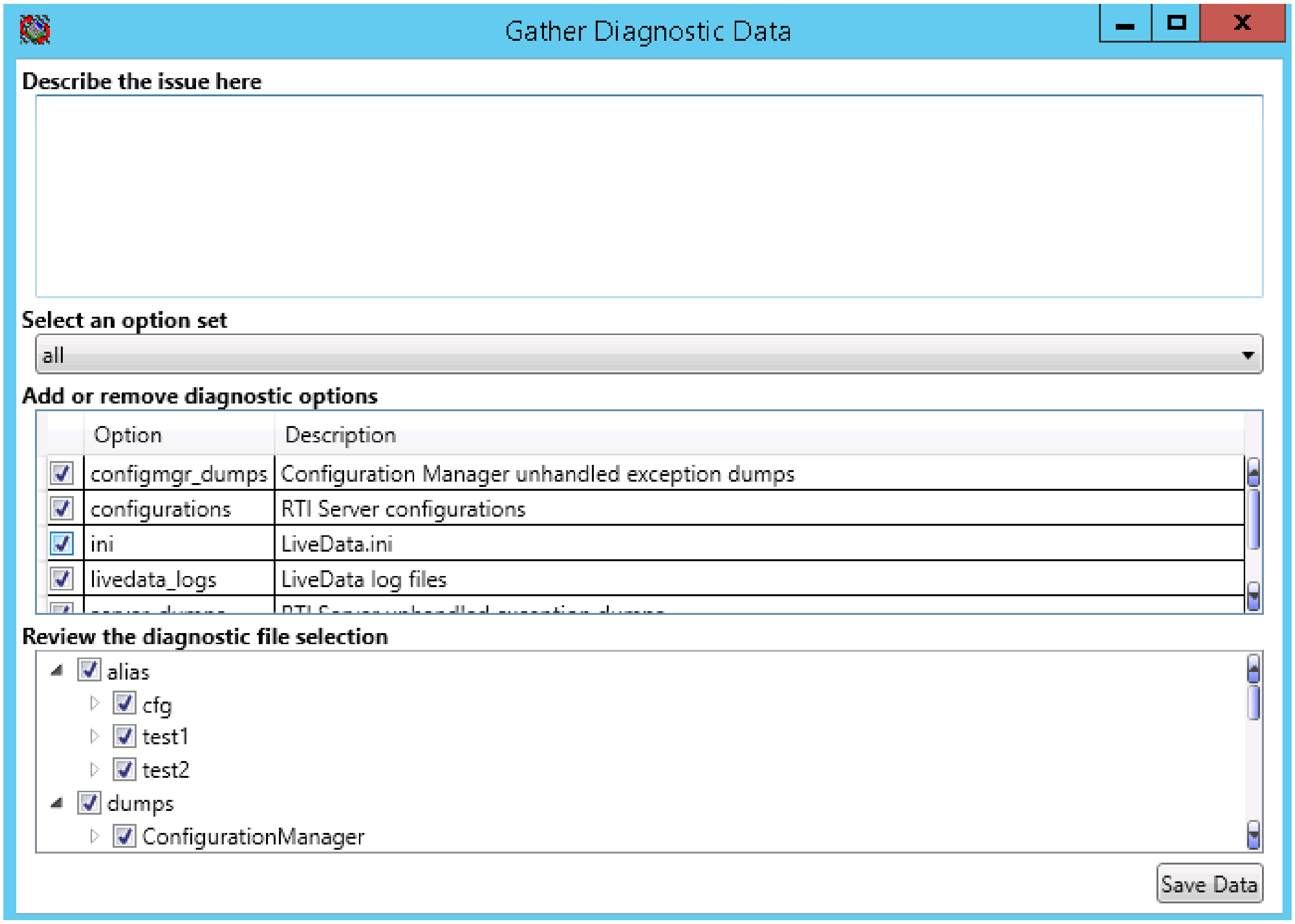
Help

The Help menu currently has two selections: **Gather Diagnostic Data** and **About.**



Gather Diagnostic Data

If you select **Gather Diagnostic Data...**, RTI Configuration Manager will display a form in which you can select all or some of the configuration, initialization, log, and dump files to have compressed in a .zip file, which you can send to LiveData Utilities Professional Services if you need to report an issue.


Gather Diagnostic Data

Describe the issue here

Select an option set

all

Add or remove diagnostic options

	Option	Description
☑	configmgr_dumps	Configuration Manager unhandled exception dumps
☑	configurations	RTI Server configurations
☑	ini	LiveData.ini
☑	livedata_logs	LiveData log files
☑	server_dumps	RTI Server unhandled exception dumps

Review the diagnostic file selection

- ☑ alias
  - ☑ cfg
  - ☑ test1
  - ☑ test2
- ☑ dumps
  - ☑ ConfigurationManager

Figure 25: Gather Diagnostic Data Form

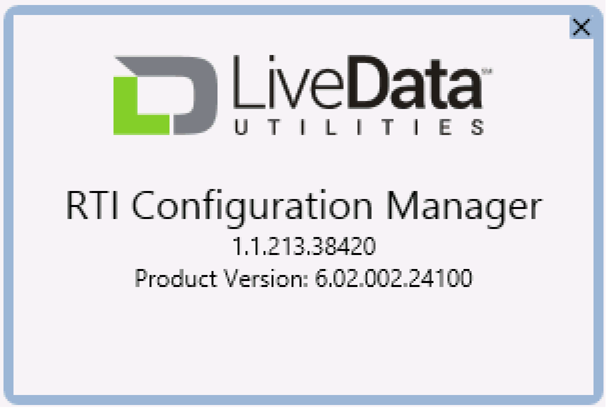
The form contains four labeled sections:

- **Describe the issue here.** Always provide a description of the issue.
- **Select an option set.** This section provides a drop-down list of sets of files that you would like to send to LiveData Utilities Professional Services: all, dump_files_only, or log_files_only. Clicking on <reset> deselects a previous choice.
- **Add or remove diagnostic options.** This section allows you to add or remove specific files.
- **Review the diagnostic file selection.** This section allows you to review the files that you have selected and deselect any that you think are not necessary.

After reviewing your selections, click on the **Save Data** button to generate, name, and save the compressed .zip file. After generating the .zip file, you can send it to LiveData Utilities Professional Services.

About

If you select **About**, RTI Configuration Manager will show the version number of RTI Configuration Manager. Figure 26 shows how it will be displayed.



RTI Configuration Manager
1.1.213.38420
Product Version: 6.02.002.24100

Figure 26: RTI Configuration Manager's About Panel

Connectors

Figure 27 shows the different types of connectors that are used within the Node Table. You can click on a connector and drag it between the nodes that you would like to connect.

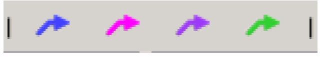


Figure 27: Connectors on the Command Bar

Update Based Connector (edge) subscribes to notifications from the source node. For storage type nodes, a write to the node will notify the connector. For input type nodes, the notify operation, if supported, notifies the connector. Otherwise, a write operator from the external device notifies the connector.

Time Based connector (edge) is designed to poll for data periodically by reading from the source node. The poll rate is determined by the poll class number. The SetPollClass macro may be used to set the poll class rate.

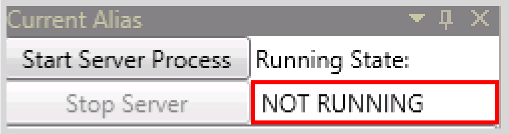
Demand Based Connector (edge) reads from its source node whenever the destination node reads from the connector. Thus, it perpetuates a read operation. For example, if a time based connector reads from an Add node, then the Add node will read from each of its inputs, and Demand based connectors would be needed at both inputs.

TwoWay Refresh Based Connector (edge) is used in special cases where data moves in both directions. For example, the association control nodes pass association status in one direction, and association control information in the other direction.

Start and Stop RTI Server

Usually, you will start and stop RTI Server as a Windows Service from the Command Bar with the Start service cfg and Stop service cfg buttons. The name of the alias is appended to the **Start Service** and **Stop Service** buttons; in this case, cfg is the alias. Microsoft Windows services enable you to create long-running executable applications that run in their own Windows sessions. In a production environment, you will always start RTI Server in this way.

An alternative to starting RTI Server is to start it as a process, but you would never start RTI Server as a process once you are in production. Generally, the **Start Server Process** button is used by LiveData Utilities Professional Services to diagnose problems or test configurations. If you start RTI Server as a process to test your changes, Windows Services Manager does not manage the application as it would if you started RTI Server as a service. Specifically, RTI Server does not start automatically when the system reboots.



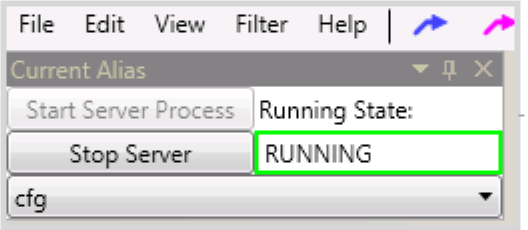
Running State:
NOT RUNNING

Figure 28: Click on the Start Server Process button to start RTI as a process

The **Stop Server** button beneath the **Start Server Process** button will stop RTI Server if you started it as a process.

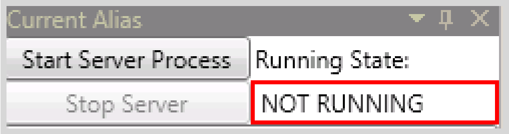
Server State

The Running State of RTI Server is displayed in the Current Alias panel as shown below.



Running State: RUNNING

Figure 29: If the state is RUNNING, the textbox is highlighted with green

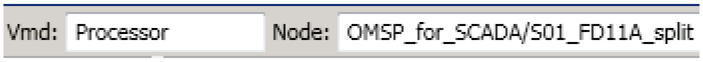


Running State: NOT RUNNING

Figure 30: If the state is NOT RUNNING, the textbox is highlighted with red

VMD and Node

Figure 31 shows the VMD and node selected from the Node Table on the Command Bar in the VMD and Node fields.



Vmd: Processor Node: OMSP_for_SCADA/S01_FD11A_split

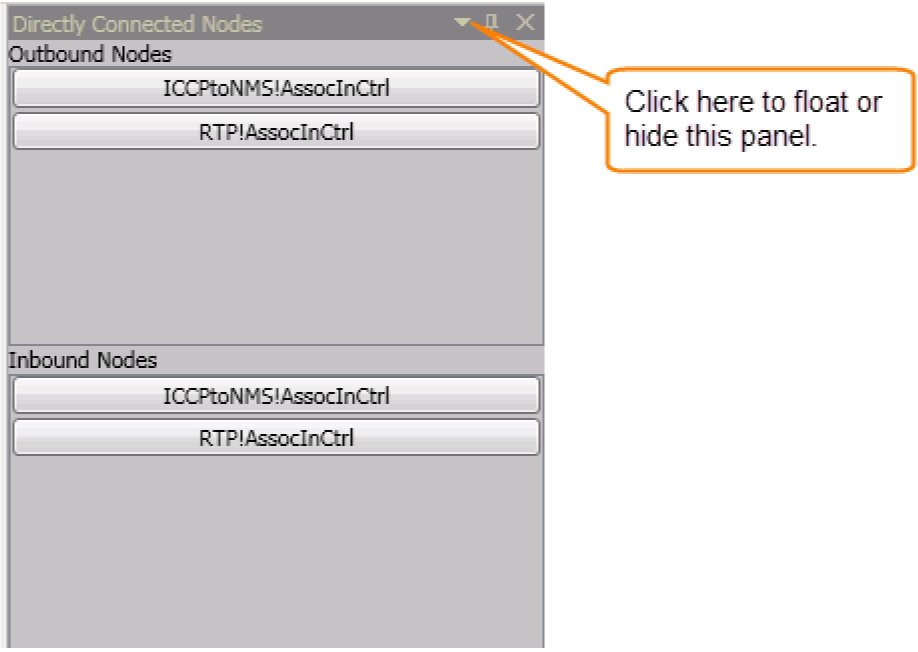
Figure 31: Vmd and Node Fields on the Command Bar

Node Table and Connectors

You can select a node by clicking on the node in the Node Table. Once you have selected a node, the VMD of that node type and the name of the node are shown in the VMD and Node fields, respectively. Also, a contrasting color behind the node highlights the node as shown in Figure 33.

When you select a node from the Node Table, you can see:

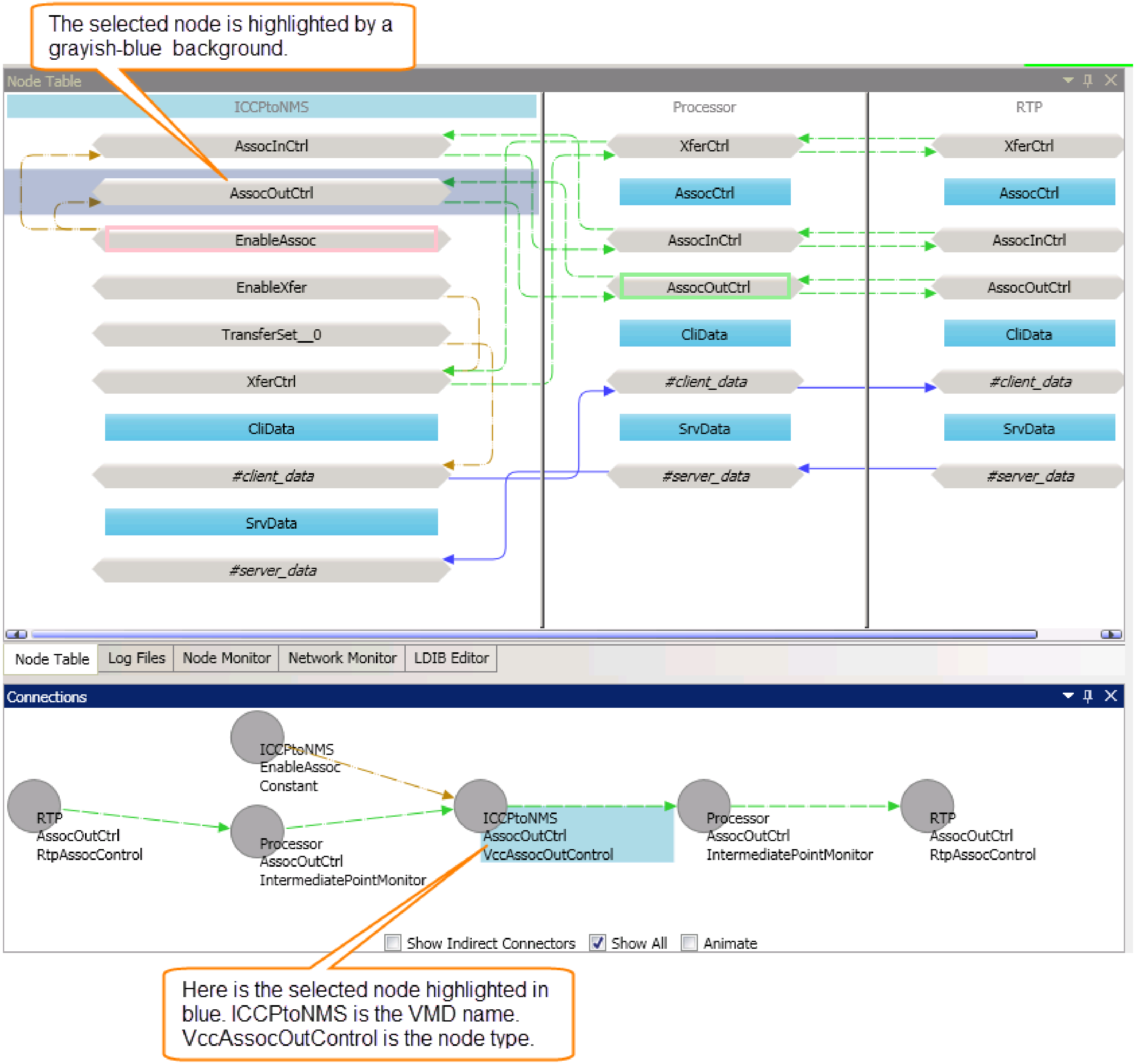
- All the connections to and from the selected node in the Connections panel directly below the Node Table.
- Directly connected nodes in the Directly Connected Nodes panel to the right of the Node Table. Outbound Nodes are shown in the top subsection and Inbound Nodes are shown in the bottom subsection as displayed in Figure 32.



Click here to float or hide this panel.

Figure 32: Directly Connected Node Panel

Figure 33 shows all of the direct connections to and from the node AssocOutCtrl that is selected in the Node Table, and Figure 34 shows all of the indirect connections to this node. Note that the node is highlighted in blue in the Connections panel and given a grayish-blue background in the Node Table.



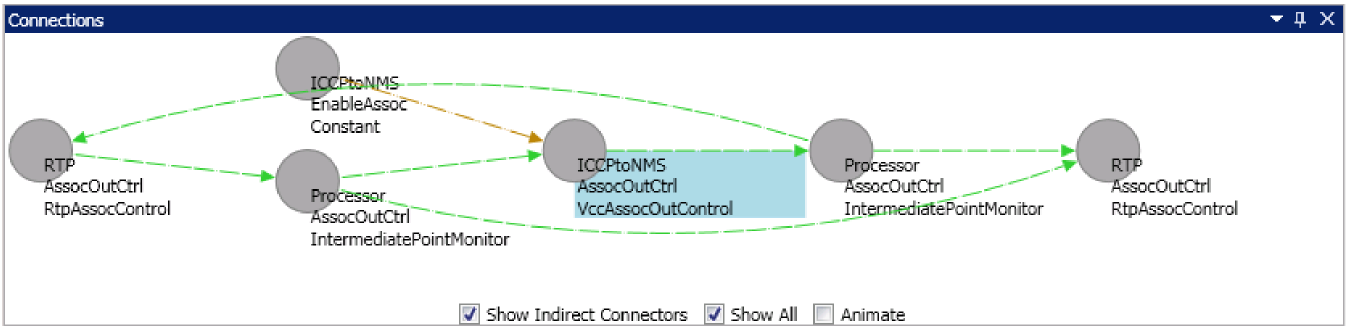
☐ Show Indirect Connectors ☑ Show All ☐ Animate

The selected node is highlighted by a grayish-blue background.

Here is the selected node highlighted in blue. ICCPtoNMS is the VMD name. VccAssocOutControl is the node type.

Figure 33: AssocOutControl Node is Selected with Direct Connectors Displayed in the Connections panel

To see the Indirect connectors in addition to the direct connectors shown in Figure 33, check **Show Indirect Connections** at the bottom of the Connections panel.


Connections
ICCPtoNMS EnableAssoc Constant
RTP AssocOutCtrl RtpAssocControl
Processor AssocOutCtrl IntermediatePointMonitor
ICCPtoNMS AssocOutCtrl VccAssocOutControl
Processor AssocOutCtrl IntermediatePointMonitor
RTP AssocOutCtrl RtpAssocControl

☑ Show Indirect Connectors ☑ Show All ☐ Animate

Figure 34: AssocOutControl Node is Selected with Direct and Indirect Connectors Displayed

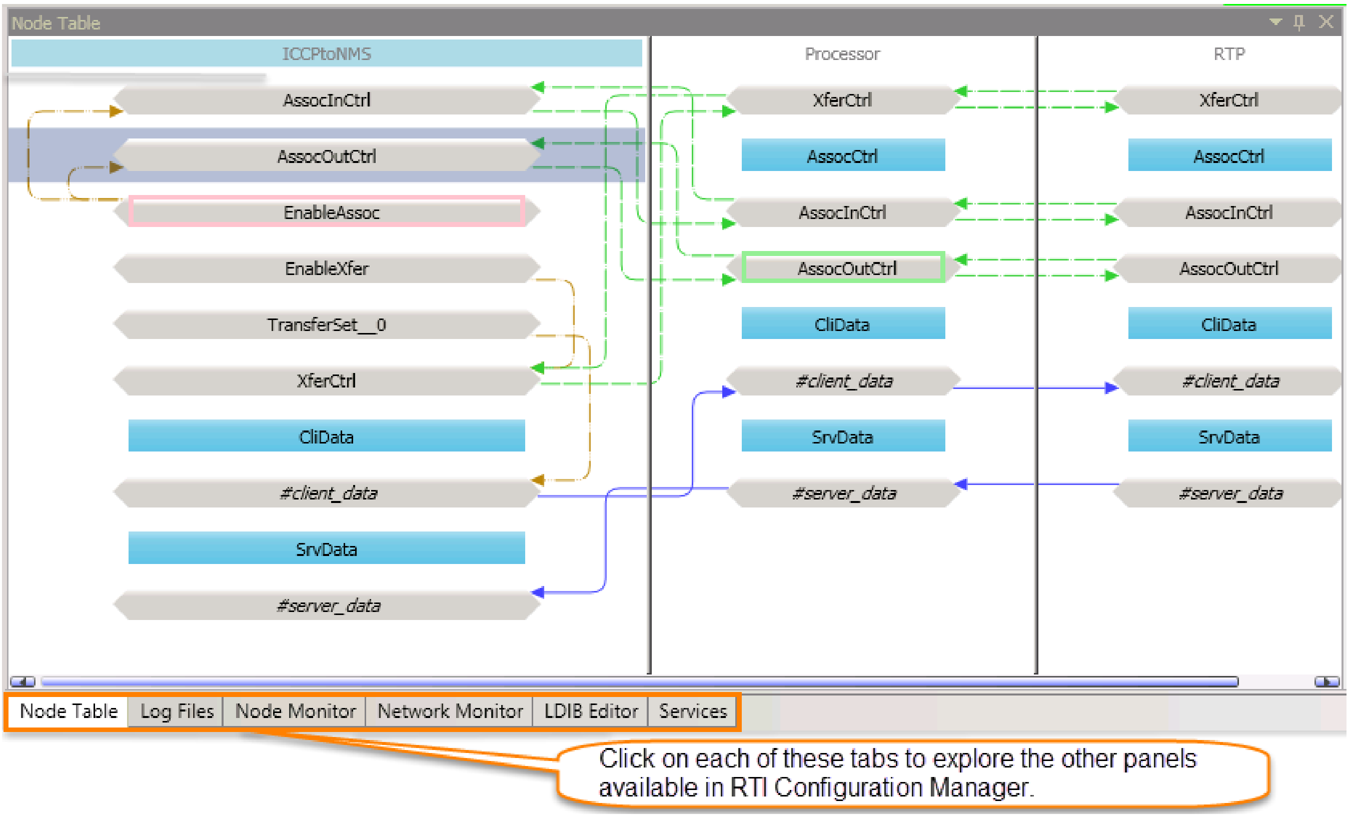
Note: Whenever you or LiveData Utilities Professional Services makes changes to the nodes or connectors in the Node Table, you need to reload the batch file or files. For more information, see Batch Files on page 50.

Tabs Below the Node Table

Figure 35 outlines the tabs underneath the Node Table in orange.

- Node Table
- Log Files
- Node Monitor
- Network Monitor
- LDIB Editor
- Services

You can click on each of these tabs to debug, monitor, or make changes to your configuration.

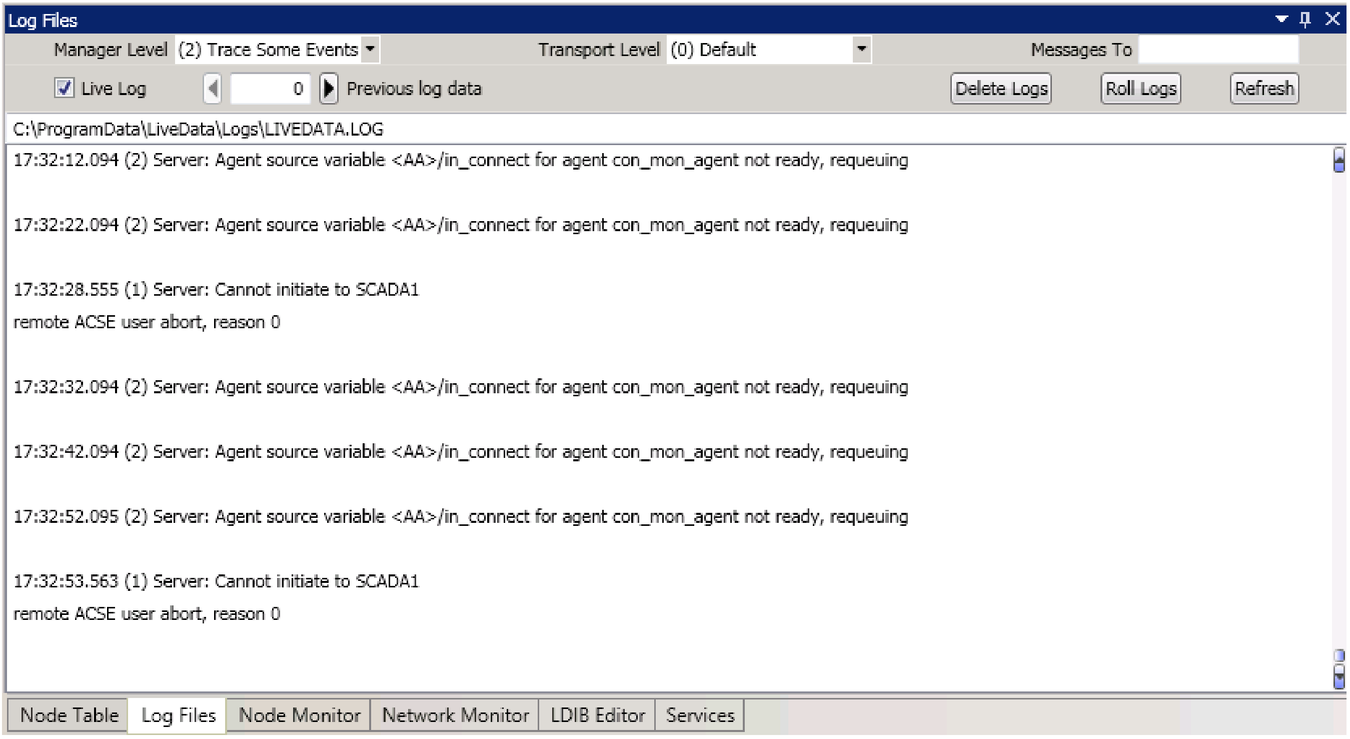


Click on each of these tabs to explore the other panels available in RTI Configuration Manager.

Figure 35: Tabs below the Node Table Show Additional panels

Log Files

The Log Files tab shows a running log of the activity in your configuration as shown in Figure 36.

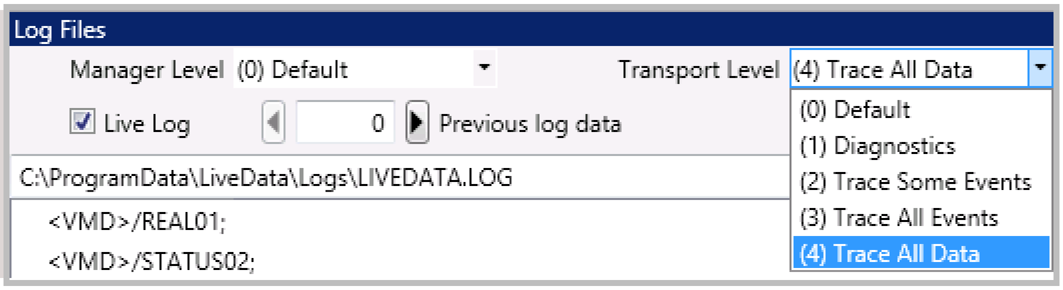


```
C:\ProgramData\LiveData\Logs\LIVEDATA.LOG
17:32:12.094 (2) Server: Agent source variable <AA>/in_connect for agent con_mon_agent not ready, requeuing

17:32:22.094 (2) Server: Agent source variable <AA>/in_connect for agent con_mon_agent not ready, requeuing

17:32:28.555 (1) Server: Cannot initiate to SCADA1
remote ACSE user abort, reason 0

17:32:32.094 (2) Server: Agent source variable <AA>/in_connect for agent con_mon_agent not ready, requeuing

17:32:42.094 (2) Server: Agent source variable <AA>/in_connect for agent con_mon_agent not ready, requeuing

17:32:52.095 (2) Server: Agent source variable <AA>/in_connect for agent con_mon_agent not ready, requeuing

17:32:53.563 (1) Server: Cannot initiate to SCADA1
remote ACSE user abort, reason 0
```

Figure 36: Log Files Tab Shows a Running Log of the Activity in Your Configuration

Log file information is divided into two types: Transport and Manager.

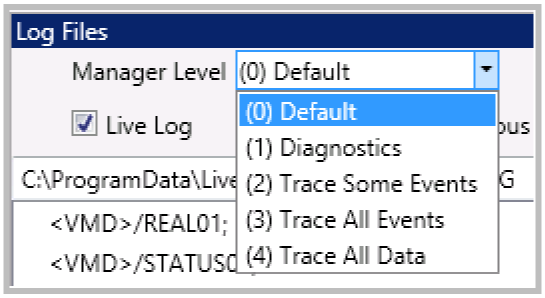
The transport layer is responsible for end-to-end communication over a network. It provides logical communication between application processes running on different hosts within a layered architecture of protocols and other network components. Transport Level logging groups information from the transport layer and other lower protocol layers. You can specify the amount of information that you would like to receive from least to most by selecting (0) Default, (1) Diagnostics, (2) Trace Some Events, (3) Trace All Events, and (4) Trace All Data from the Transport Level drop-down list (see Figure 37).



```
C:\ProgramData\LiveData\Logs\LIVEDATA.LOG
  <VMD>/REAL01;
  <VMD>/STATUS02;
```

Figure 37: Transport Level

The manager level groups information from the upper protocol layers: Session, Presentation, and ACSE. You can specify the amount of information that you would like to receive from least to most by selecting (0) Default, (1) Diagnostics, (2) Trace Some Events, (3) Trace All Events, and (4) Trace All Data from the Manager Level drop-down list (see Figure 38).



Manager Level (0) Default

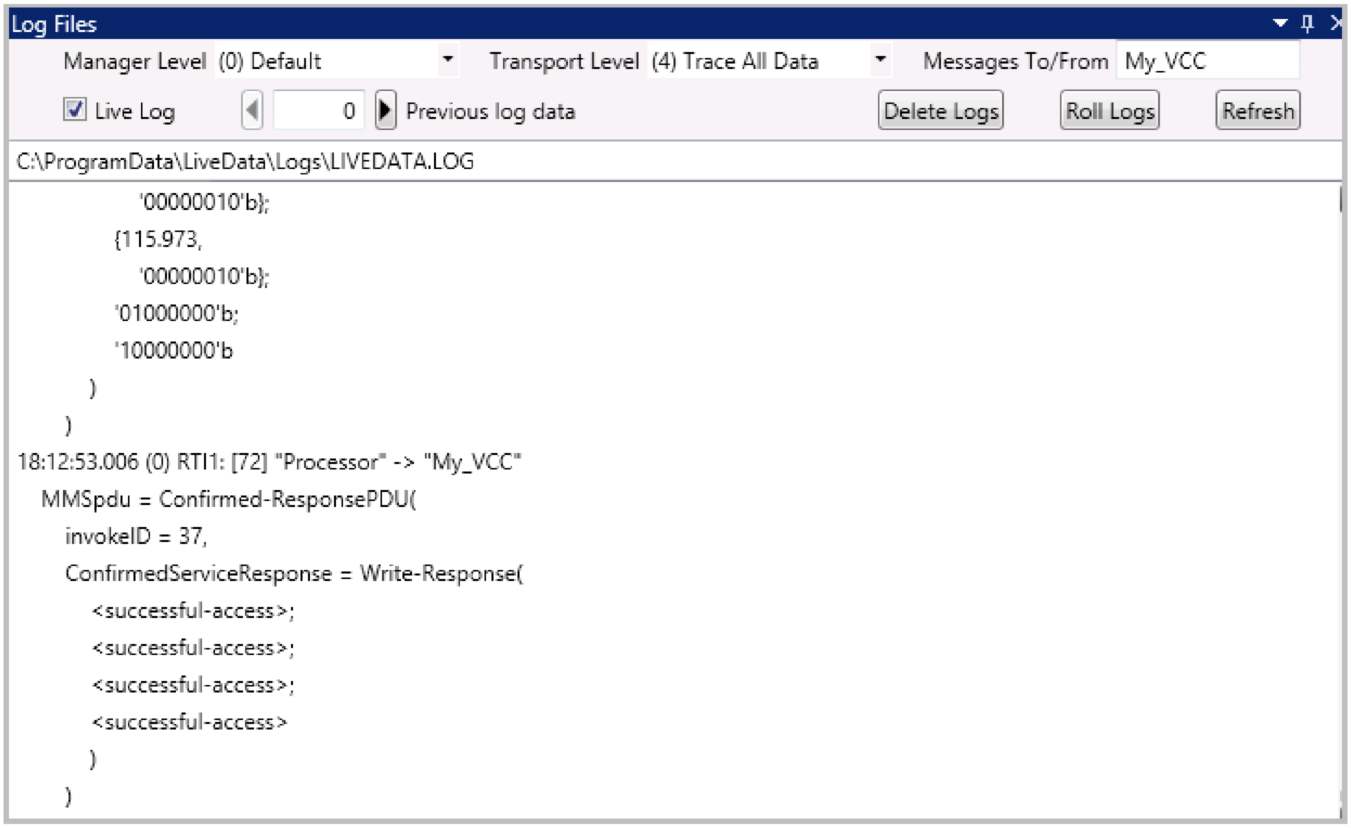
☑ Live Log

```
C:\ProgramData\Live
  <VMD>/REAL01;
  <VMD>/STATUSC
```

Figure 38: Manager Level Drop-Down List

Pinpoint the Information that is Logged

You can pinpoint the information that you need to see by specifying a VMD within the network in the **Messages To/From** field.



Manager Level (0) Default Transport Level (4) Trace All Data Messages To/From My_VCC

☑ Live Log 0 Previous log data

```
C:\ProgramData\LiveData\Logs\LIVEDATA.LOG
          '00000010'b};
        {115.973,
          '00000010'b};
        '01000000'b;
        '10000000'b
      )
    )
18:12:53.006 (0) RTI1: [72] "Processor" -> "My_VCC"
  MMSpdu = Confirmed-ResponsePDU(
    invokeID = 37,
    ConfirmedServiceResponse = Write-Response(
      <successful-access>;
      <successful-access>;
      <successful-access>;
      <successful-access>
      )
    )
```

Figure 39: Messages To/From My_VCC

Stop and Start the Live Log

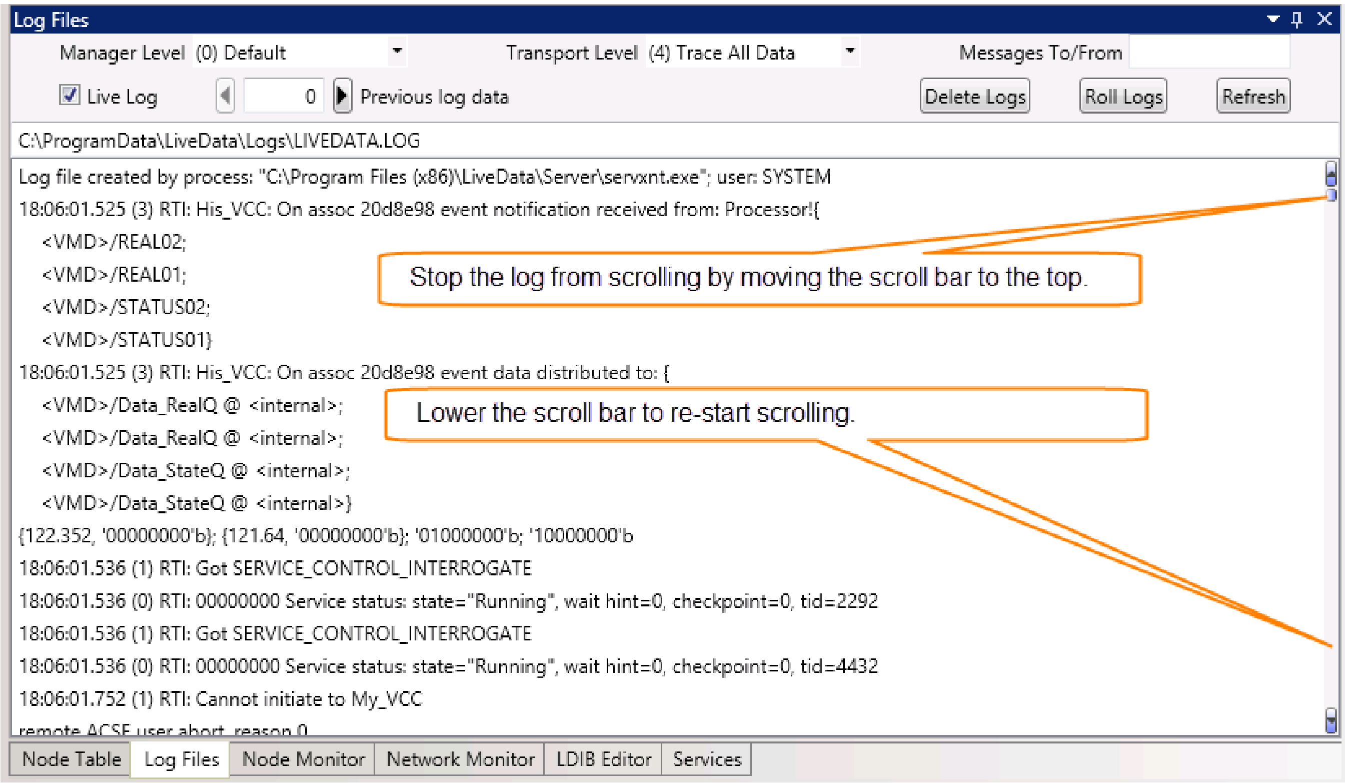
You can stop and start the Live Log by unchecking the checkbox next to Live Log label, which is below the Manager Level label in Figure 38.

Scroll through the Live Log

Events and data are shown in reverse chronological order starting with the most recent at the top of the log. You can use the scroll bar to the right side of the panel to see past events on your log.

Stop the Log from Scrolling

Sometimes the log is scrolling too fast for you to examine it. Move the scroll bar to the top of the screen to stop the log from scrolling.



```
Log file created by process: "C:\Program Files (x86)\LiveData\Server\servxnt.exe"; user: SYSTEM
18:06:01.525 (3) RTI: His_VCC: On assoc 20d8e98 event notification received from: Processor!{
    <VMD>/REAL02;
    <VMD>/REAL01;
    <VMD>/STATUS02;
    <VMD>/STATUS01}
18:06:01.525 (3) RTI: His_VCC: On assoc 20d8e98 event data distributed to: {
    <VMD>/Data_RealQ @ <internal>;
    <VMD>/Data_RealQ @ <internal>;
    <VMD>/Data_StateQ @ <internal>;
    <VMD>/Data_StateQ @ <internal>}
{122.352, '00000000'b}; {121.64, '00000000'b}; '01000000'b; '10000000'b
18:06:01.536 (1) RTI: Got SERVICE_CONTROL_INTERROGATE
18:06:01.536 (0) RTI: 00000000 Service status: state="Running", wait hint=0, checkpoint=0, tid=2292
18:06:01.536 (1) RTI: Got SERVICE_CONTROL_INTERROGATE
18:06:01.536 (0) RTI: 00000000 Service status: state="Running", wait hint=0, checkpoint=0, tid=4432
18:06:01.752 (1) RTI: Cannot initiate to My_VCC
remote ACSE user abort, reason 0
```

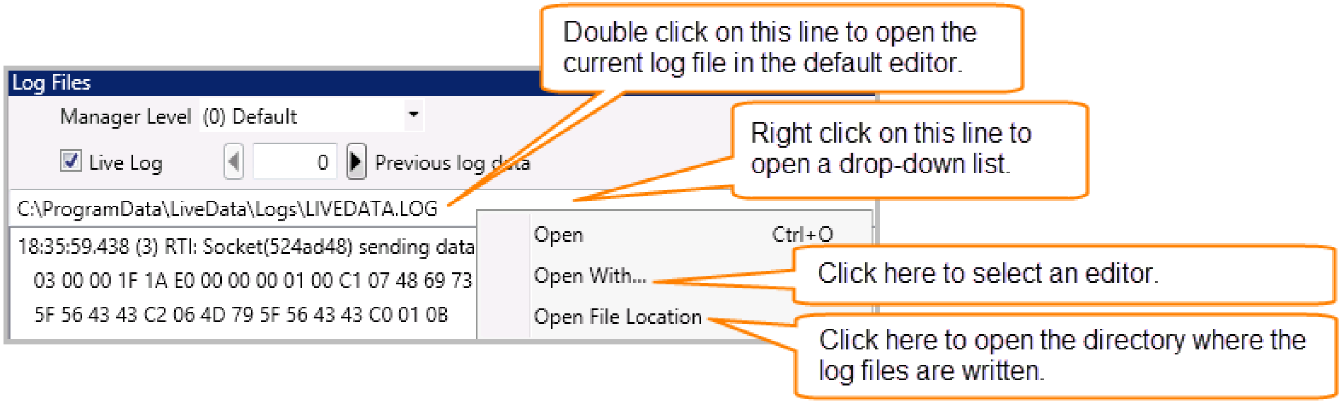
Stop the log from scrolling by moving the scroll bar to the top.

Lower the scroll bar to re-start scrolling.

Figure 40: Moving the Scroll Bar to the Top Stops the Log from Scrolling

Edit the Log File

You can open a log file in an editor, usually Notepad. Figure 41 shows the different ways that you can open a file for editing.



Manager Level (0) Default

☑ Live Log 0 Previous log data

```
C:\ProgramData\LiveData\Logs\LIVEDATA.LOG
18:35:59.438 (3) RTI: Socket(524ad48) sending data
  03 00 00 1F 1A E0 00 00 00 01 00 C1 07 48 69 73
  5F 56 43 43 C2 06 4D 79 5F 56 43 43 C0 01 0B
```

Double click on this line to open the current log file in the default editor.

Right click on this line to open a drop-down list.

Click here to select an editor.

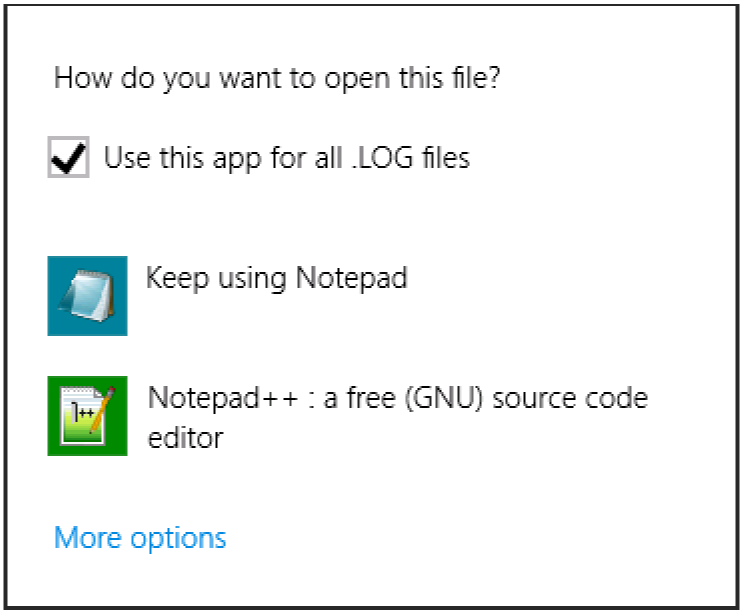
Click here to open the directory where the log files are written.

Figure 41: Editing a Log File

Note: When you open a log file, RTI Server stops writing to the file unless you use an editor that allows RTI Server to continue writing to the file after you have opened it for inspection. One such editor is Notepad++. If you would like to use an editor other than Notepad, you first must ensure that the editor is installed on your system.

To select an alternative editor:

1. Right click on the log file's pathname and select **Open With...** from the drop-down list.



How do you want to open this file?

☑ Use this app for all .LOG files

Keep using Notepad

Notepad++ : a free (GNU) source code editor

More options

Figure 42: Select Your Default Editor

2. Click on the editor that you would like to use. If it does not appear in the list, click on **More options**. If the editor that you would like to use still does not appear, click on the link Look for another app on this PC at the end of the list to search for your editor of choice.

Delete Logs

You can delete all log files and start with a new log by clicking on the Delete Logs button on the top right of the Log Files panel. While debugging a configuration, it is helpful to stop the service, delete the existing log files, and restart the service to see if there are any issues when the configuration first starts.

Roll Logs

You can roll the contents of the active log in LIVEDATA.LOG to another file named LD_00001.LOG by clicking on the Roll Logs button in the top right corner of the Log Files panel.

The contents that were in LD_00001.LOG would be rolled into the file named LD_00002.LOG. The number of log files that you can have is configurable as is the size of a log file. For more information, speak to LiveData Utilities Professional Services.

Refresh Logs

If the active log has been paused, you can refresh it by clicking on the Refresh button in the right corner of the Log Files panel.

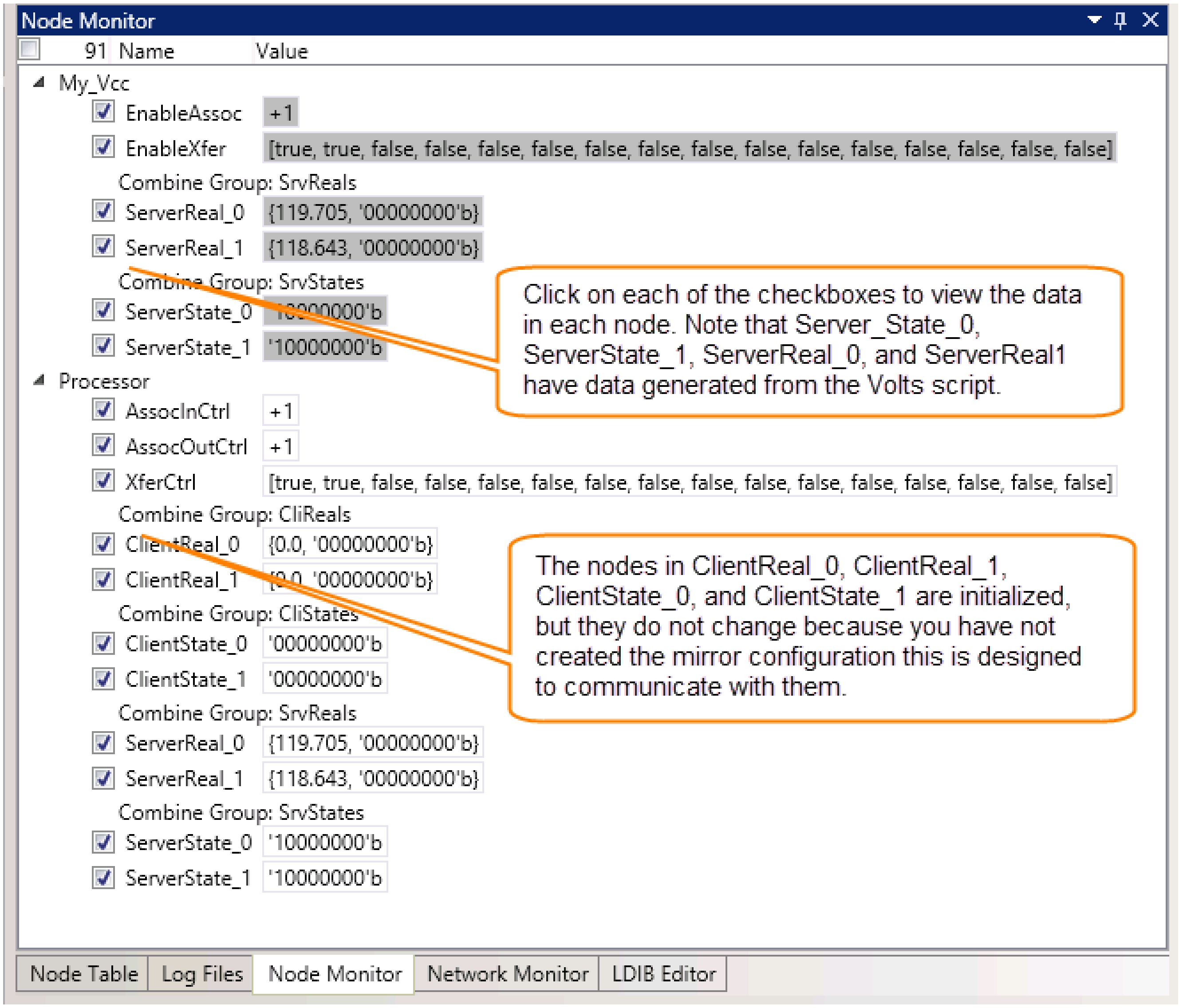
Node Monitor

After you have a working configuration, you can start the configuration and examine the contents of each node in the Node Monitor.

To examine the nodes in your configuration:

1. Click on the Node Monitor tab at the bottom of the Node Table.
2. Click on **Start service** in the RTI Configuration Manager header if RTI Server is not already started.
3. Click on each of the checkboxes next to the monitorable nodes that you would like to see.
4. Notice the values changing in the monitorable nodes on the display.

Figure 43 shows an example of the Node Monitor displaying the values of each monitorable node in the configuration.



□ 91	Name	Value
◢ My_Vcc		
☑	EnableAssoc	+1
☑	EnableXfer	[true, true, false, false, false, false, false, false, false, false, false, false, false, false, false, false]
	Combine Group: SrvReals	
☑	ServerReal_0	{119.705, '00000000'b}
☑	ServerReal_1	{118.643, '00000000'b}
	Combine Group: SrvStates	
☑	ServerState_0	'10000000'b
☑	ServerState_1	'10000000'b
◢ Processor		
☑	AssocInCtrl	+1
☑	AssocOutCtrl	+1
☑	XferCtrl	[true, true, false, false, false, false, false, false, false, false, false, false, false, false, false, false]
	Combine Group: CliReals	
☑	ClientReal_0	{0.0, '00000000'b}
☑	ClientReal_1	{0.0, '00000000'b}
	Combine Group: CliStates	
☑	ClientState_0	'00000000'b
☑	ClientState_1	'00000000'b
	Combine Group: SrvReals	
☑	ServerReal_0	{119.705, '00000000'b}
☑	ServerReal_1	{118.643, '00000000'b}
	Combine Group: SrvStates	
☑	ServerState_0	'10000000'b
☑	ServerState_1	'10000000'b

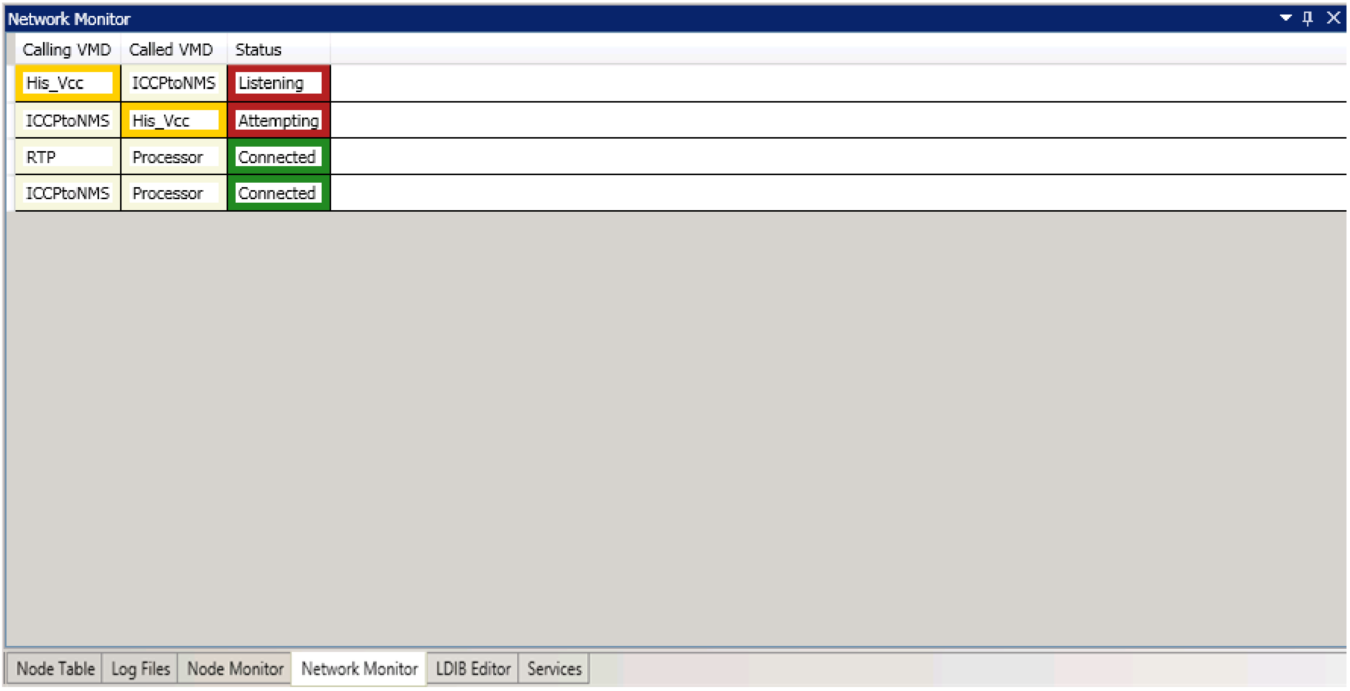
Click on each of the checkboxes to view the data in each node. Note that Server_State_0, ServerState_1, ServerReal_0, and ServerReal1 have data generated from the Volts script.

The nodes in ClientReal_0, ClientReal_1, ClientState_0, and ClientState_1 are initialized, but they do not change because you have not created the mirror configuration this is designed to communicate with them.

Figure 43: Start RTI Server and View the Data in Each Node

Network Monitor

Figure 44 shows the Network Monitor panel.



Calling VMD	Called VMD	Status
His_Vcc	ICCPtoNMS	Listening
ICCPtoNMS	His_Vcc	Attempting
RTP	Processor	Connected
ICCPtoNMS	Processor	Connected

Figure 44: Network Monitor Tab Shows the Status of each VMD Connection

The Network Monitor panel shows MMS association statuses: Listening, Attempting, Connected, and Disabled.

Listening means the device is waiting for an inbound connection attempt.

Attempting means that the device is making outbound connection attempts.

Connected means that the two devices are connected.

Disabled means that the device's ability to connect has been turned off.

To display a legend of the different types of endpoint VMDs:

- Click **View>Legend** from the Command Bar.
- Check **Show network status monitor legend** from the list.

Figure 45 shows a legend so that you will be able to discern the types of devices.

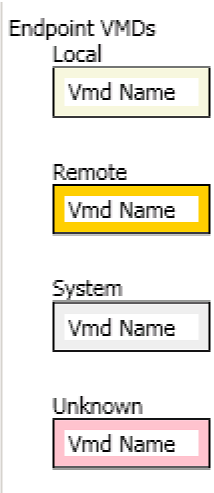
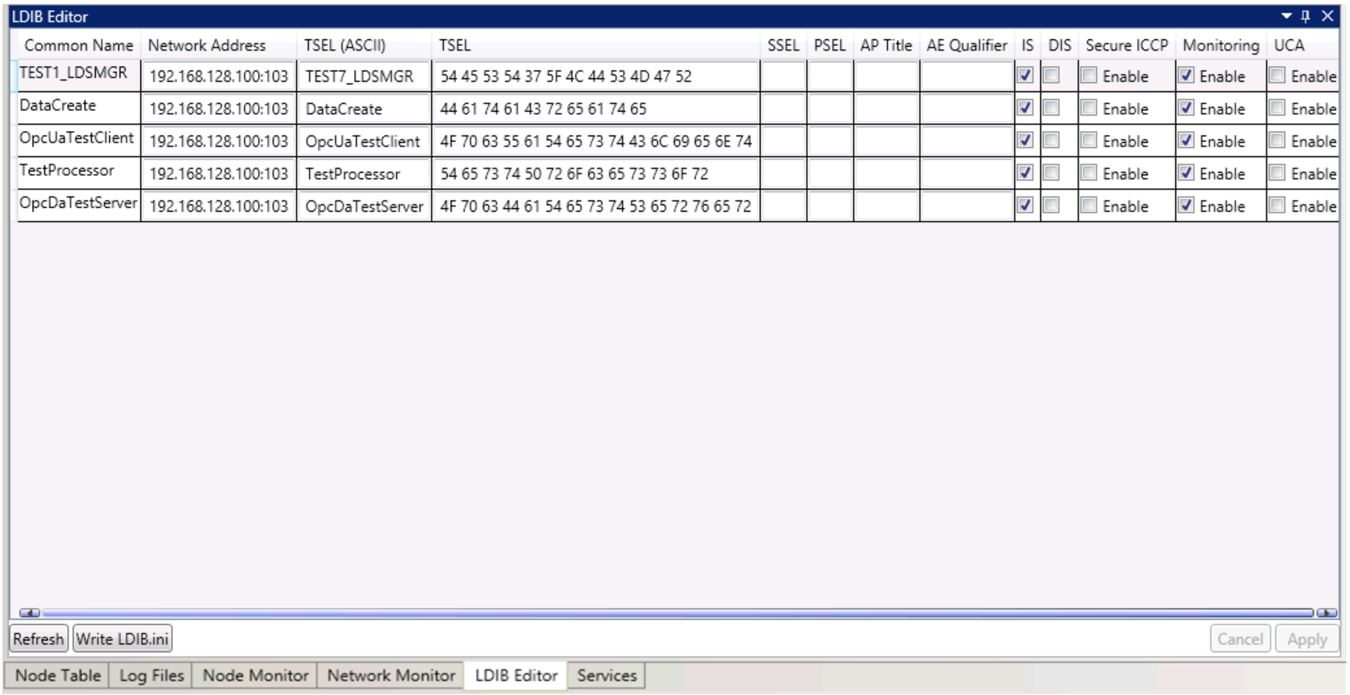


Figure 45: Legend of the Endpoint VMDs that are in the Network Monitor

LDIB Editor

Figure 46 shows the LDIB Editor. It allows you to check the network addresses and other parameters and make any changes if necessary.



Common Name	Network Address	TSEL (ASCII)	TSEL	SSEL	PSEL	AP Title	AE Qualifier	IS	DIS	Secure ICCP	Monitoring	UCA
TEST1_LDSMGR	192.168.128.100:103	TEST7_LDSMGR	54 45 53 54 37 5F 4C 44 53 4D 47 52					☑	☐	☐ Enable	☑ Enable	☐ Enable
DataCreate	192.168.128.100:103	DataCreate	44 61 74 61 43 72 65 61 74 65					☑	☐	☐ Enable	☑ Enable	☐ Enable
OpcUaTestClient	192.168.128.100:103	OpcUaTestClient	4F 70 63 55 61 54 65 73 74 43 6C 69 65 6E 74					☑	☐	☐ Enable	☑ Enable	☐ Enable
TestProcessor	192.168.128.100:103	TestProcessor	54 65 73 74 50 72 6F 63 65 73 73 6F 72					☑	☐	☐ Enable	☑ Enable	☐ Enable
OpcDaTestServer	192.168.128.100:103	OpcDaTestServer	4F 70 63 44 61 54 65 73 74 53 65 72 76 65 72					☑	☐	☐ Enable	☑ Enable	☐ Enable

Figure 46: LDIB Editor

If you have loaded a new or changed batch file, click on the **Refresh** button in order to see the current values.

If you need to change the IP address of a VMD, you can change it in the Network Address column and then click on the **Apply** button.

If you suspect that your ICCP addressing parameters are incorrect, you can check under the TSEL, SSEL, PSEL, AP Title, and AE Qualifier columns to see what they are set to in your configuration, make any necessary changes, and apply your changes by clicking on the **Apply** button.

Note: These changes will override values that you set if you or someone else loaded one or more batch files. However, you can restore these values if you reload the original batch file(s).

If you do not specify a TSEL address, RTI Configuration Manager spells out the type of device under TSEL (ASCII) and the hexadecimal equivalent of the name under TSEL, for example, **TestProcessor** and **54 65 73 74 50 72 6F 63 65 73 73 6F 72** as shown in the first row in Figure 46.

The additional columns are explained below.

- IS - Internation Standard is mutually exclusive with DIS. Select either IS or DIS.
- DIS - Draft International Standard is mutually exclusive with IS. Select either IS or DIS.
- Secure ICCP - Uncheck the box in order to disable Secure ICCP.
- Monitoring - Check the box to enable monitoring. Uncheck it to disable monitoring.
- UCA - Check the box in order to disable the Utility Communications Architecture.

LDIB for LiveData MMS Client Control or LiveData Client Manage Variables Applet

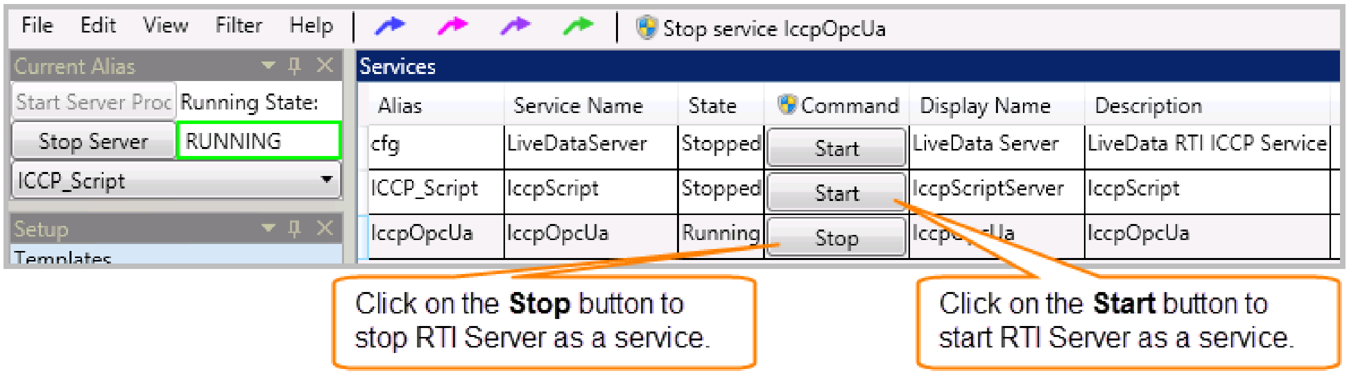
If your configuration needs to communicate with a LiveData MMS Client Control (ActiveX Control) program or LiveData Client Manage Variables Applet, and you change an address or any of the networking parameters of a VMD in LDIB, you need to use the **Write LDIB.ini** button. Clicking on the **Write LDIB.ini** button exports your VMD network addressing records into the requisite INI file format, which LiveData MMS Client Control and LiveData Client Manage Variables Applet require to resolve VMD names to network addresses. Note that you **do not** have to click on this button for changes to prototype nodes, nodes, or connectors.

Secondarily, you can click on the **Write LDIB.ini** button to generate the LDIB.ini file, which allows you to validate your configured VMD addressing information.

Note: Clicking on the **Write LDIB.ini** button will replace the existing LDIB.ini with one that matches the LDIB for the current configuration. Existing entries in LDIB.ini will be overwritten.

Services

From the Services panel, you can stop and start RTI Server as a Windows service for multiple configurations. In this way, you can monitor how two or more separate configurations communicate when they are active.



Alias	Service Name	State	Command	Display Name	Description
cfg	LiveDataServer	Stopped	Start	LiveData Server	LiveData RTI ICCP Service
ICCP_Script	IccpScript	Stopped	Start	IccpScriptServer	IccpScript
IccpOpcUa	IccpOpcUa	Running	Stop	IccpOpcUa	IccpOpcUa

Click on the **Stop** button to stop RTI Server as a service.

Click on the **Start** button to start RTI Server as a service.

Figure 47: Services Panel

To create a service for an alias, see the section File on page 23

To delete a service for an alias, see the section Edit on page 25.

Tabs Below the Properties Panel

There are five tabs below the Properties panel. Each of these tabs has a different purpose.

- Server
- VMD
- Node
- Connector
- Batch Files
    - Create a new batch file
    - Load or reload a batch file
    - Load part of a batch file
    - Unload a batch file

Batch Files

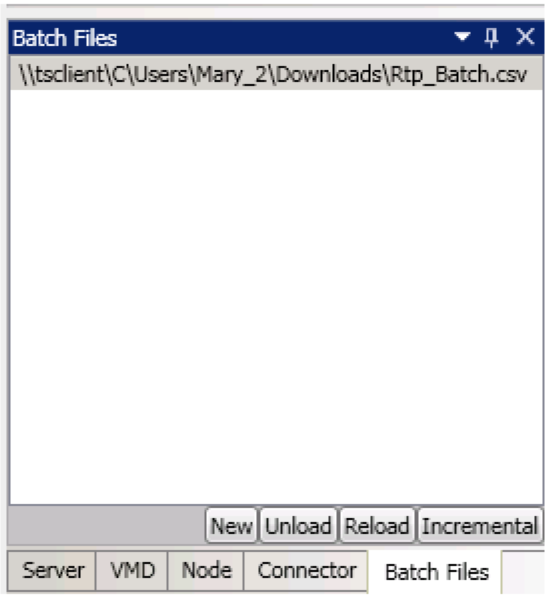


Figure 48: Batch Files Panel

To create a new batch file

- Click on the **New** button to create a new batch file. This will open the LiveData\Config directory in Windows Explorer. RTI Configuration Manager, by default, names the file New Batch.csv.
- Press the **Open** button to edit it in Notepad.
- Save the file with **File>Save As** to give the file an appropriate name.
- Stop RTI Server by clicking **Stop Server** from the Command Bar.
- After the server has stopped in Windows Task Manager, select the **Load Share** or **Load No Share** depending on how you would like to load the batch file.
- Restart RTI Server by clicking **Start Server Service**.

To reload a modified batch file

Reloading brings in the existing batch file if you have modified it or if you have changed the configuration (in the Node Table) since you last loaded the file.

- Stop RTI Server by clicking **Stop Server** from the Command Bar.
- Click on the **Reload** button to bring in the revised batch file.

- Restart RTI Server by clicking **Start Server Service**.

If you have not changed the configuration in the Node Table and you are not loading more than one batch file, you do not need to stop RTI Server.

To load only the parts of the batch file that have changed

The following procedure saves time if you need to load a particularly large batch file.

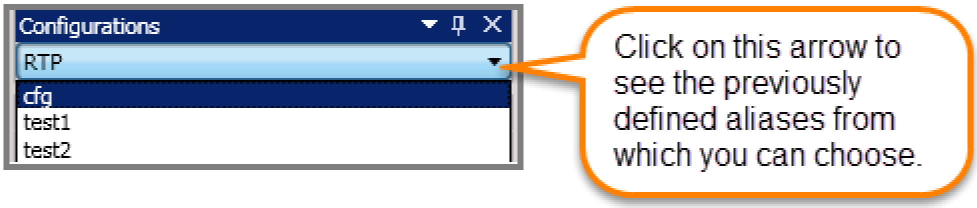
- Stop RTI Server by clicking on **Stop Server** from the Command Bar.
- Click on the **Incremental** button.
- Restart RTI Server by clicking on **Start Server Service**.

To unload a batch file

- Select the batch file from the Batch Files panel.
- Click on the **Unload** button to unload it.

Alias Selection and Creation

Figure 49 shows the panel from where you can select a different alias. An alias is an internal buffer. You can import a configuration that is defined in a .db file into any of the aliases. You can also create new aliases with descriptive names for your configuration; however, most people import their production configuration into "cfg."



Click on this arrow to see the previously defined aliases from which you can choose.

Figure 49: Select a Different Alias

When you first install RTI Configuration Manager, it comes with the following empty aliases that you can use as a starting point from which to import and modify configurations supplied by :

- cfg
- test1
- test2

To select a different alias:

1. Choose an alias from the drop-down menu shown in "Select a Different Alias" above

To create a new alias:

1. Click on **New alias...** from the File menu as shown in "Create a New Alias from the File Menu" below

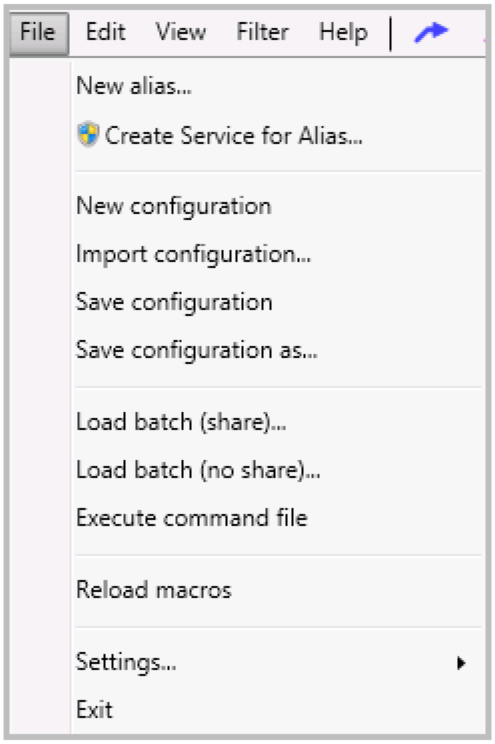
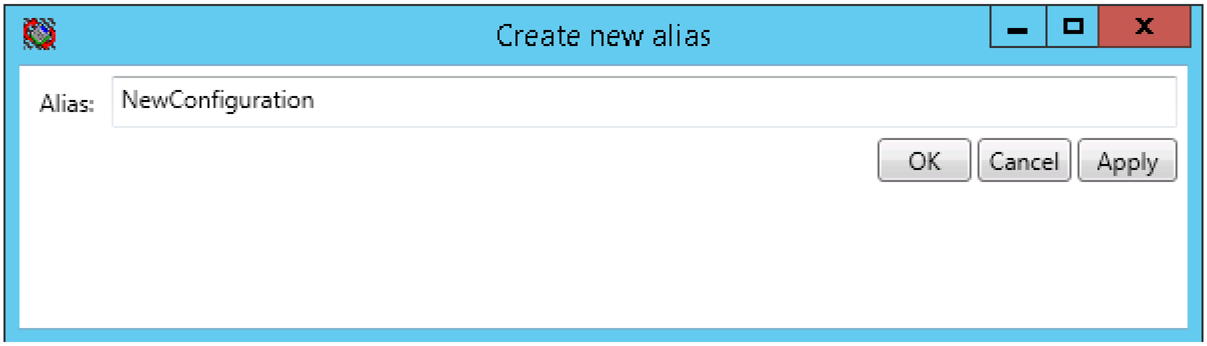


Figure 50: Create a New Alias from the File Menu

2. RTI Configuration Manager displays the Create new alias form.



3. Delete "NewConfiguration" from the text box and type in a name for the new alias.

4. Click **OK** to create one alias and leave the Create new alias form, or you can click on the **Apply** button if you intend to create one or more additional aliases and do not want to leave the form.

Monitor Your Dataflow

RTI Configuration Manager allows you to monitor all the nodes and connections in your dataflow with the following tools:

- Node Monitor
  - Where you can watch the values of all monitorable nodes change in a running configuration.
  - For more information, see Node Monitor on page 45.
- Variable Access
  - Where you can monitor the value of a specified variable in a running configuration at any given point in time.
  - For more information, see Variable Access on page 28.
- Network Monitor
  - Where you can see the Connection status, indicating if you are connected to any of the external systems or not.
  - For more information, see Network Monitor on page 47.
- LDIB Editor
  - Where you can view and modify connection parameters, such as IP addresses and various layer selectors:
    - Network Service Access Point or NSAP (IP Address)
    - Transport Selector (TSEL)
    - Session Selector (SSEL)
    - Presentation Selector (PSEL)
  - For more information, see LDIB Editor on page 48.

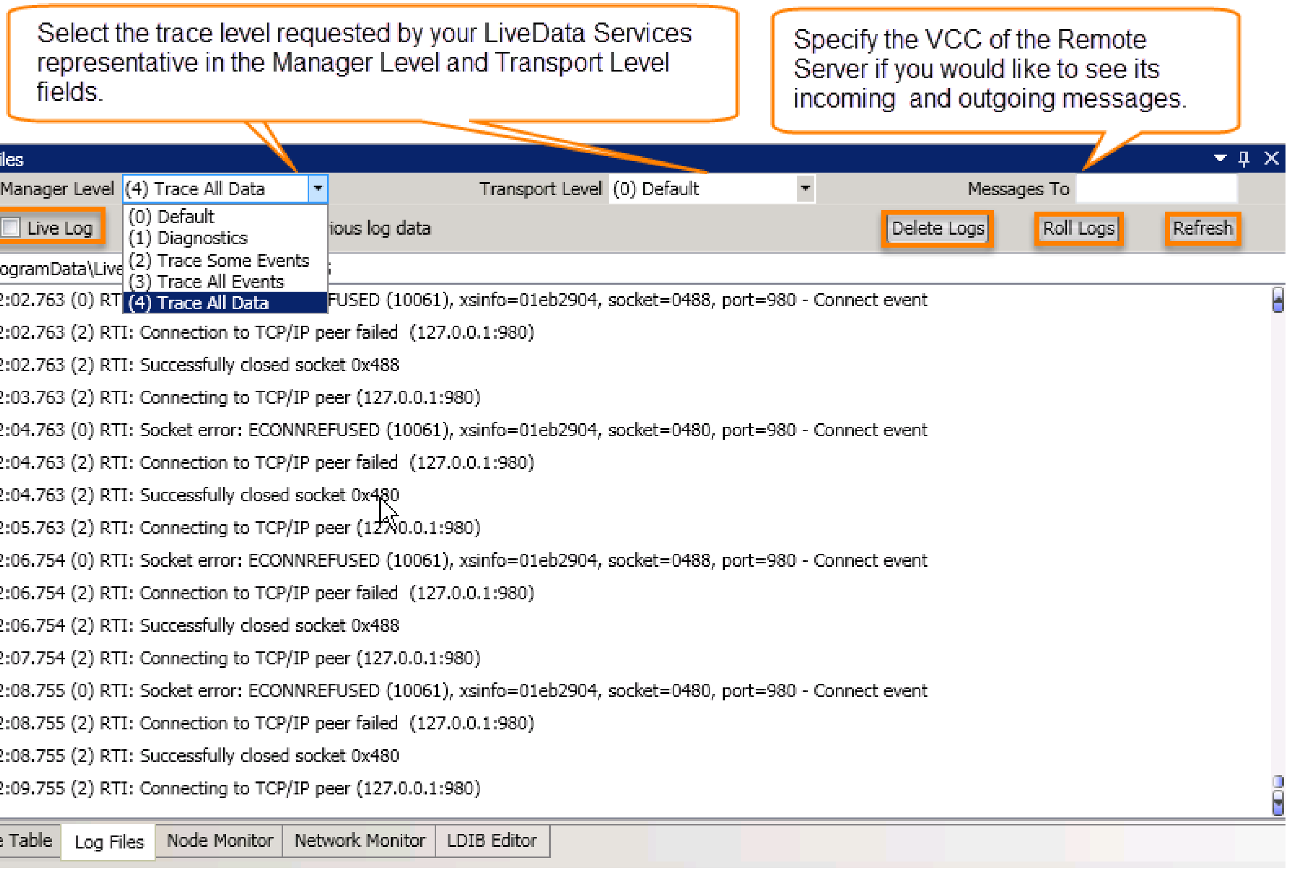
Advanced Monitoring and Debugging Tools

RTI Configuration Manager provides the following monitoring and debugging resources for advanced users:

- Log Files, which is where you can adjust logging levels. It is also where you can View, Delete or Roll the log files.
- Server Status panel, which is where you can view a host of server performance metrics. This panel is mostly for debugging purposes and is designed for more advanced users and LiveData Utilities Professional Services engineers.

Log Files

Figure 51 shows the Log Files panel and provides some description of the different fields, highlighting the Live Log checkbox and various buttons with an orange outline.



Select the trace level requested by your LiveData Services representative in the Manager Level and Transport Level fields.

Specify the VCC of the Remote Server if you would like to see its incoming and outgoing messages.

```
14:12:02.763 (0) RT                    FUSED (10061), xsinfo=01eb2904, socket=0488, port=980 - Connect event
14:12:02.763 (2) RTI: Connection to TCP/IP peer failed  (127.0.0.1:980)
14:12:02.763 (2) RTI: Successfully closed socket 0x488
14:12:03.763 (2) RTI: Connecting to TCP/IP peer (127.0.0.1:980)
14:12:04.763 (0) RTI: Socket error: ECONNREFUSED (10061), xsinfo=01eb2904, socket=0480, port=980 - Connect event
14:12:04.763 (2) RTI: Connection to TCP/IP peer failed  (127.0.0.1:980)
14:12:04.763 (2) RTI: Successfully closed socket 0x480
14:12:05.763 (2) RTI: Connecting to TCP/IP peer (127.0.0.1:980)
14:12:06.754 (0) RTI: Socket error: ECONNREFUSED (10061), xsinfo=01eb2904, socket=0488, port=980 - Connect event
14:12:06.754 (2) RTI: Connection to TCP/IP peer failed  (127.0.0.1:980)
14:12:06.754 (2) RTI: Successfully closed socket 0x488
14:12:07.754 (2) RTI: Connecting to TCP/IP peer (127.0.0.1:980)
14:12:08.755 (0) RTI: Socket error: ECONNREFUSED (10061), xsinfo=01eb2904, socket=0480, port=980 - Connect event
14:12:08.755 (2) RTI: Connection to TCP/IP peer failed  (127.0.0.1:980)
14:12:08.755 (2) RTI: Successfully closed socket 0x480
14:12:09.755 (2) RTI: Connecting to TCP/IP peer (127.0.0.1:980)
```

Figure 51: Log Files panel with Annotation

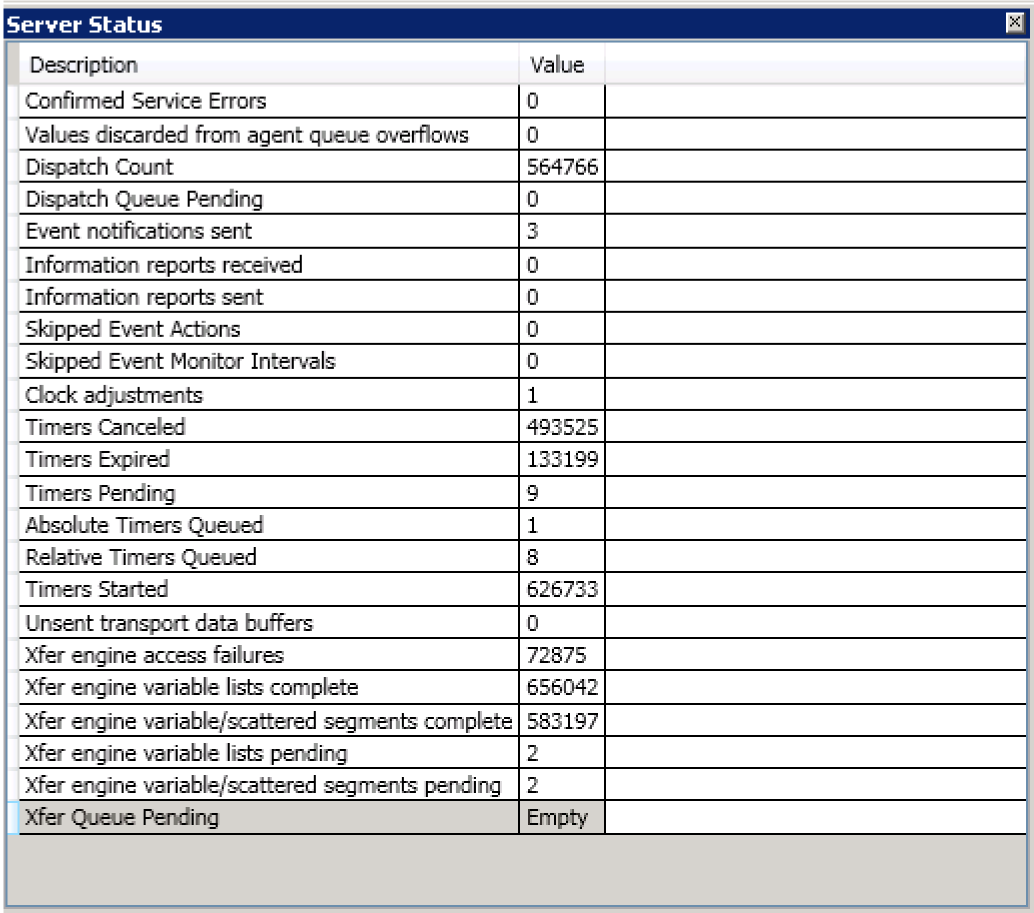
LiveData Utilities Professional Services will often ask you to adjust logging levels in order to debug a configuration or find a solution to a request you might have. Use the drop-down lists from the Manager Level and Transport Level fields to specify more or less information in the log as shown in Figure 51.

From the top of this panel, you can do any of the following things:

- Switch on a real-time log by checking or unchecking (to turn it off) the Live Log checkbox outlined in orange in Figure 51.
- Click on the Delete Logs button (outlined in orange in Figure 51) to delete all existing log files.
- Click on the Roll Logs button (outlined in orange in Figure 51) to save the current log file and restart logging if you would like to perform a test during a specific period of time, and would like RTI Configuration Manager to turn off the current log and start fresh at the beginning of the test period.
- Click the Refresh button (outlined in orange in Figure 51) to refresh the log if you have not checked the Live Log checkbox.

Server Status Panel

Figure 52 shows RTI Server Status panel.



Description	Value
Confirmed Service Errors	0
Values discarded from agent queue overflows	0
Dispatch Count	564766
Dispatch Queue Pending	0
Event notifications sent	3
Information reports received	0
Information reports sent	0
Skipped Event Actions	0
Skipped Event Monitor Intervals	0
Clock adjustments	1
Timers Canceled	493525
Timers Expired	133199
Timers Pending	9
Absolute Timers Queued	1
Relative Timers Queued	8
Timers Started	626733
Unsent transport data buffers	0
Xfer engine access failures	72875
Xfer engine variable lists complete	656042
Xfer engine variable/scattered segments complete	583197
Xfer engine variable lists pending	2
Xfer engine variable/scattered segments pending	2
Xfer Queue Pending	Empty

Figure 52: Server Status panel

In a running configuration, you can see the performance metrics change in the RTI Server Status panel.

To see this panel:

- Click on View from the Command Bar.
- Select panels.
- Check Server status.

This will display RTI Server Status panel shown in Figure 52.

Note: You cannot change these metrics. These metrics are provided by RTI Configuration Manager for you to examine how RTI Server is performing at any given time. See your LiveData Utilities Professional Services engineer for more information.

RTI Configuration Manager Tutorial

RTI Configuration Manager allows you to implement a configuration in a number of ways:

- You can import an existing template for the type of configuration that you intend to implement. Then modify and add to them. These templates have descriptive names, such as ICCPModbusTemplate.db, DnpMasterScriptTemplate.db, and OpcUaServerTemplate.db, and are located in D: Program Data\LiveData. You will need to know how to change the parameters of VMDs, create new VMDs, change the properties of prototype nodes, nodes , and connectors, and make new ones. Finally, you will need to create and load batch files.
- If you are an experienced user, you can create a configuration .db on your own. You will need to have a thorough understanding of how to create a configuration and batch files. You will also need to know how to use the protocol parameters associated with all protocols that you intend to use. In addition, you will need to know how to create new VMDs, prototype nodes, nodes , and connectors.
- If you are new to RTI Configuration Manager, you can have a LiveData Utilities Professional Services engineer provide you with a configuration .db that you must import into RTI Configuration Manager. You will need to know how to load batch files, make changes to batch files, monitor your configuration or configurations, and make changes to the configuration(s) if the need arises.

Overview: This section provides an annotated graphical representation of a configuration, which is intended to provide a conceptual understanding of how a configuration is implemented in RTI Configuration Manager. The subsequent subsections explain how to create virtual devices, prototype nodes, and nodes.

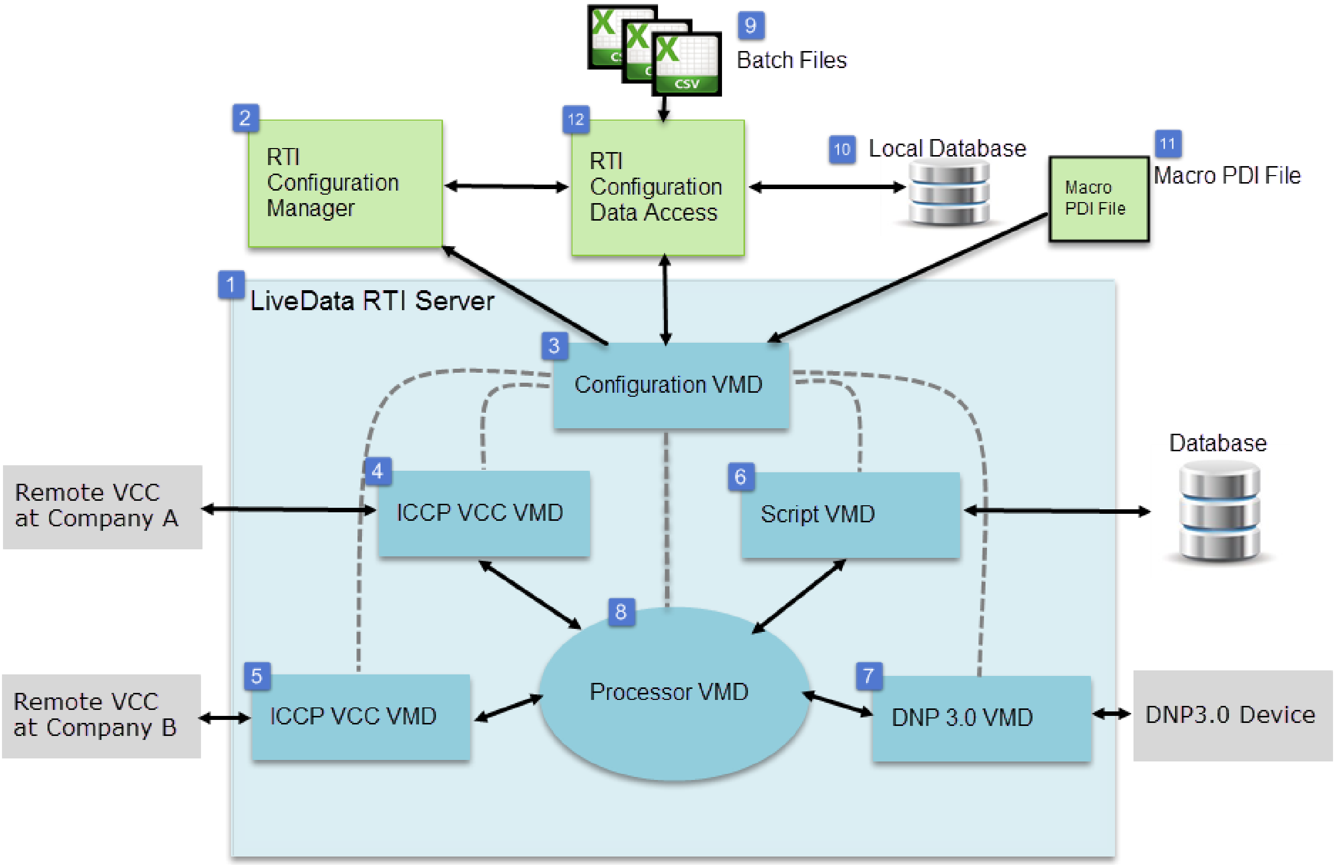
Appendix I on Implementing an ICCP Configuration: The first section provides a step-by-step procedure for creating an ICCP configuration without importing a pre-made configuration. The second section describes how you can import a pre-existing configuration .db and make adjustments to it. This section will help to familiarize you with the process of making small changes to an existing database configuration.

Appendix II on Implementing an OPC UA Configuration: This appendix describes how to import a number of pre-made configurations as you would do if a LiveData Utilities Professional Services engineer prepared them for you. It provides a guide to using OPC UA templates and various related properties.

Together the overview and tutorial provide a basis from which you can start importing templates in order to create your own configurations.

Overview

RTI Configuration Manager enables you to create a visual representation of a network. This visual representation provides an internal variable model that mirrors your electrical network in LiveData RTI Server. The following illustration shows how RTI Configuration Manager works with LiveData RTI Server to implement a network.



1 Represents one configuration among an infinite number of configurations that LiveData RTI Server can implement. In this particular configuration, there are two virtual control centers (VCCs). Each of these VCCs communicates with a remote device through the ICCP protocol. There is also a DNP3.0 VMD, which communicates with a SCADA device through the DNP 3.0 protocol. Finally, the configuration communicates with an external database through a database access script provided by RTI Configuration Manager.

2 Represents RTI Configuration Manager that allows you to create the visual representation of a network, placing each VCC, processor, script, prototype node, connector, and other objects into LiveData RTI Server as needed. Once you have created a configuration, you can load batch files that create the hundreds or thousands of actual points in a network, allowing you to trace and monitor the dataflow from any point.

3 Represents the Configuration VMD, the one element that all RTI Server configurations have in common. It places the objects created in RTI Configuration Manager into RTI Server.

4 Represents an ICCP Virtual Control Center VMD that communicates with a remote device at Company A. This is an example of one VCC that RTI Configuration Manager can configure.

5 Represents another ICCP Virtual Control Center VMD that communicates with another remote device at Company B.

6 Represents a database Script VMD. There are many different types of scripts; this one provides database access.

7 Represents a DNP 3.0 VMD that allows communication with a remote SCADA system through the DNP 3.0 protocol.

8 Represents the Processor VMD that acts as the central hub for communication among the VCCs and DNP 3.0 device in this example or for additional VMDs, such as different scripts and other protocols in other configurations. Although the Processor VMD supports no specific protocol nor does it perform any data transformation on its own, it is the common place to add filters to change incoming data before sending it out to its destination.

9 Represents batch files that create (instantiate) the hundreds or thousands of actual nodes, also known as points, using the corresponding prototype nodes defined within your configuration. These batch files are .csv files, easily created with a spreadsheet or text editor.

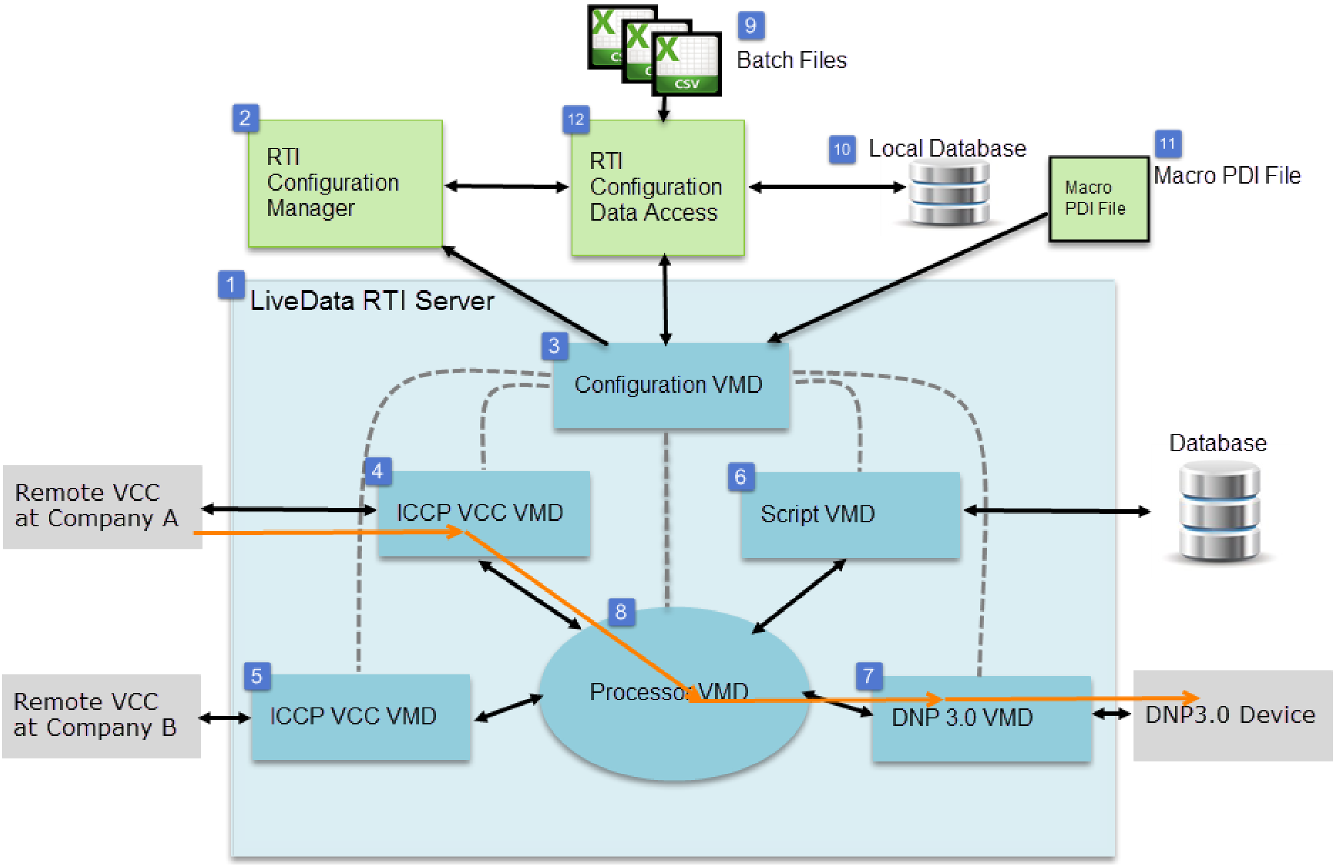
10 Represents the local file-based database that stores your entire configuration, including VCCs, scripts, prototype nodes, and actual points loaded in from one or more batch files. If you stop LiveData Server or RTI Configuration Manager, RTI Configuration Manager can bring your local database back so that you can edit your configuration and so that LiveData Server can access it.

11 The Macro file DataflowMacros.pdi contains the macros RTI Server needs to create all the different types of nodes and connectors in a configuration. You define the arguments for each node and each connector within RTI Configuration Manager and then supply an additional set of arguments from a batch file or files in order to instantiate the nodes and connectors in your entire configuration. In addition to the ICCP and DNP3 nodes in this illustration, Modbus and OPC protocol nodes are available for communicating with devices, and REST and ActiveMQ nodes are provided for communicating with web-based systems. If a new feature is added or a bug is fixed, these macros need to be updated. To reload macros from RTI Configuration Manager, you can use the Reload command from the File menu.

12 Represents the mechanism that transforms the configuration data from RTI Configuration Manager into a form that can be stored in the local database and into another form that can be executed by RTI Server. After you load a batch file, this mechanism expands the prototype configuration (represented in RTI Configuration Manager) into an executable configuration.

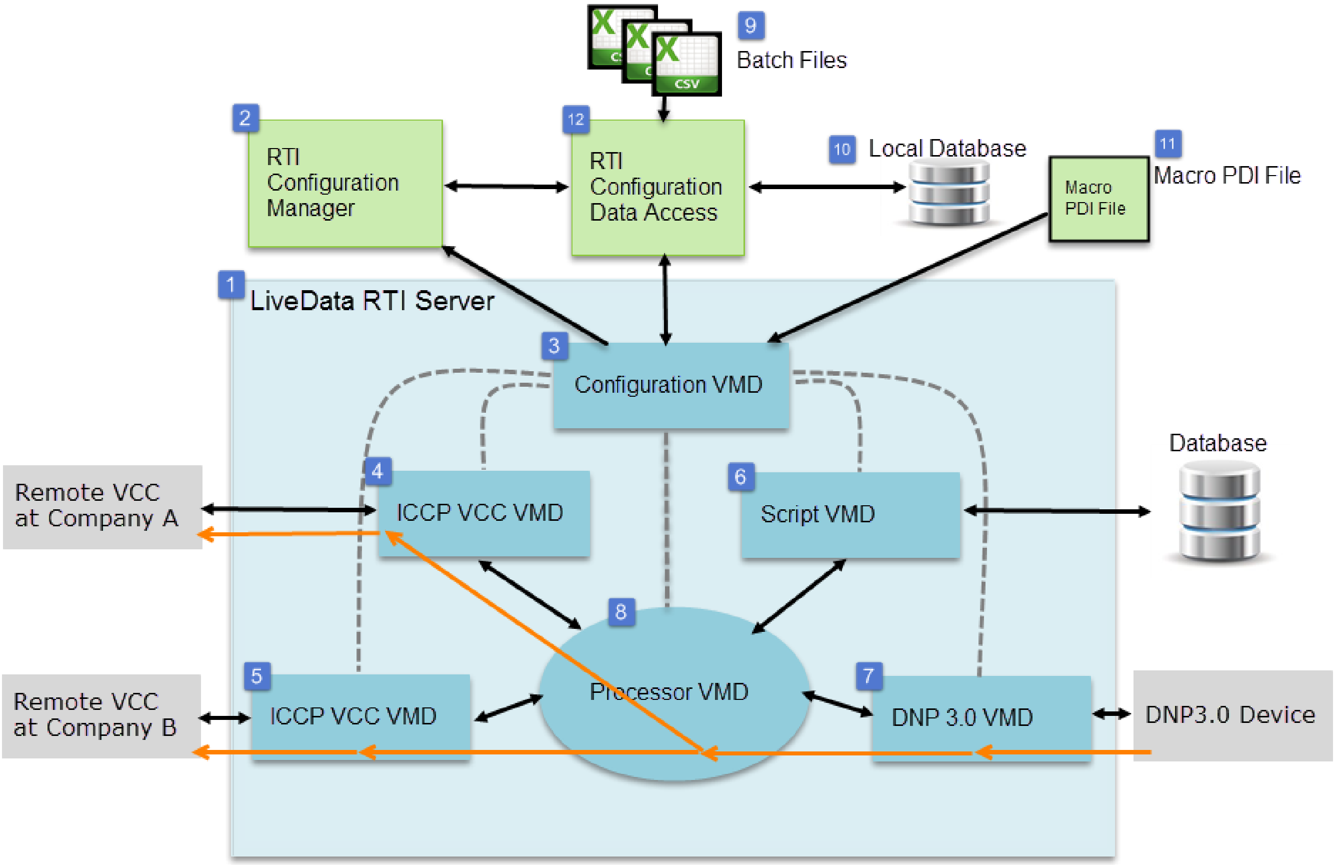
Example Dataflow 1

Data could flow from the first ICCP VCC representing a device at Company A 4 to the Processor 8 to the DNP 3.0 VMD 7 and finally to the remote DNP3.0 device.



Example Dataflow 2

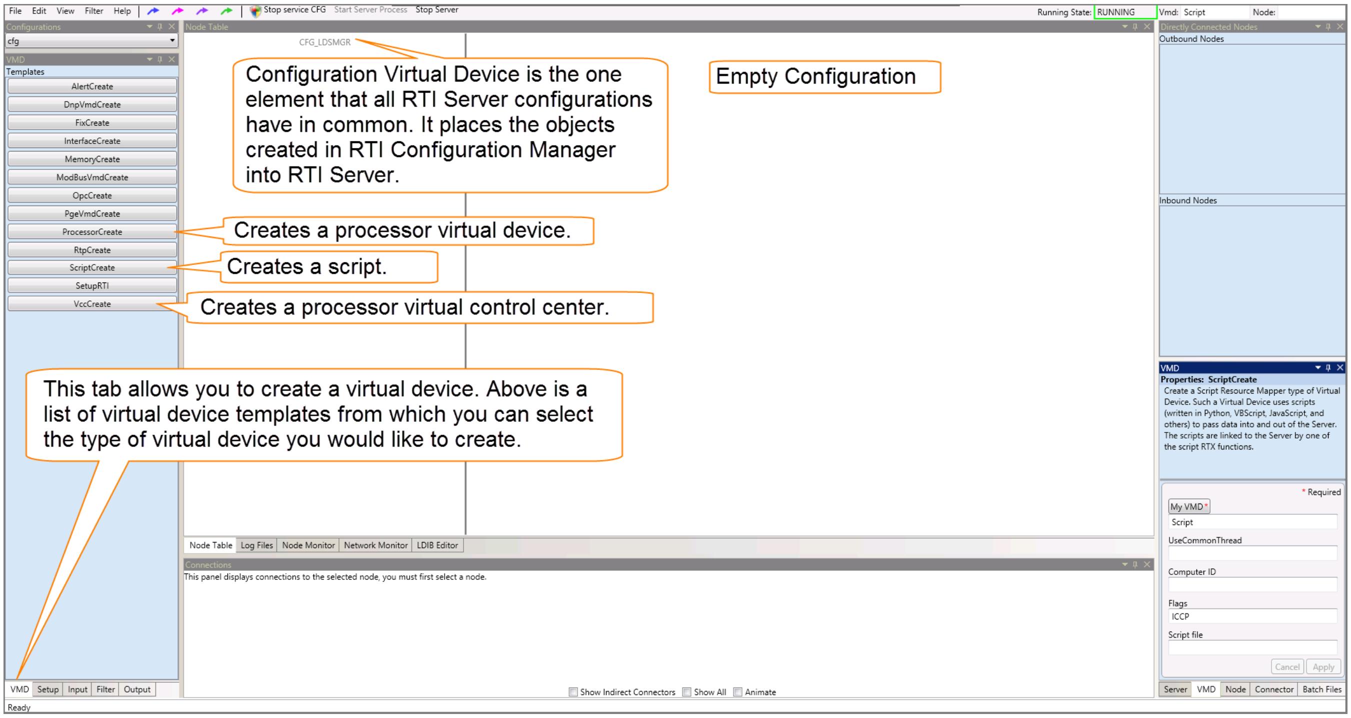
Data could also flow from a SCADA system using the DNP 3.0 protocol through the Processor to Company A and to Company B.



Data could flow from any VMD to any other VMD as long as it goes through the Processor at the center of the diagram.

Creating a Virtual Device

A LiveData Utilities Professional Services engineer will provide you with a configuration .db that you must import into RTI Configuration Manager, so you do not need to create VMDs or nodes, but you do need to know how to make changes to them if the need arises. This section gives you a general overview of how to create virtual devices and nodes because you would use the same templates and properties in order to make small adjustments to VMDs and nodes or to add nodes later on.



Configuration Virtual Device is the one element that all RTI Server configurations have in common. It places the objects created in RTI Configuration Manager into RTI Server.

Empty Configuration

Creates a processor virtual device.

Creates a script.

Creates a processor virtual control center.

This tab allows you to create a virtual device. Above is a list of virtual device templates from which you can select the type of virtual device you would like to create.

This panel displays connections to the selected node, you must first select a node.

Properties: ScriptCreate

Create a Script Resource Mapper type of Virtual Device. Such a Virtual Device uses scripts (written in Python, VBScript, JavaScript, and others) to pass data into and out of the Server. The scripts are linked to the Server by one of the script RTX functions.

* Required

My VMD *: Script

UseCommonThread:

Computer ID:

Flags: ICCP

Script file:

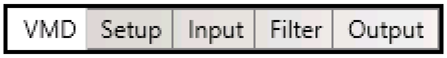
From here, you can create a virtual device (VMD) within LiveData RTI Server. Each VMD is capable of communicating with another VMD, which may be within RTI Server or external to it (possibly in a different location). A VMD is a container for other objects that you need in order to implement a network.

You must create a node within a VMD; thus, the first step in creating a new configuration is the creation of one or more VMDs to serve as containers for the nodes. Most configurations start with three VMDs.

A LiveData Utilities Professional Services engineer will provide you with a configuration .db that you must import into RTI Configuration Manager, so you do not need to create VMDs or nodes, but you do need to know how to make changes to them if the need arises.

Create Additional Types of Nodes

In addition to VMDs, you can create Setup, Input, Filter, and Output nodes by clicking on the appropriate tabs.



An **Input node**[1] sends information into RTI Server.

An **Output node**[2] sends information out from RTI Server.

A **Filter node**[3] provides a means for modifying data as it flows through the node or identify a point where you want to monitor data as it flows through the node.

A **Setup node**[4] can provide general information to RTI Server, such as poll classes and association lists; some Setup nodes are involved in controlling associations, and others provide for bulk transfer of input or output data, such as setting up a script or an ICCP transfer set to handle a group of points. In general, a Setup node does not deal with data on a point by point basis.

[1] A node that brings information into LiveData RTI Server from the external world by way of one of the protocols supported by RTI Server. There are, however, some input nodes that provide access information within RTI Server, such as a constant, system variable, or Memory VMD.

[2] An output node generally sends information out from RTI Server to the external world by way of on of the protocols supported by RTI Server. There are, however, some output nodes that provide access to information within RTI Server, such as nodes in a Memory VMD, which map onto RTI Server memory.

[3] A filter node is used in a general sense to refer to a node through which data may pass within RTI Server as opposed to Input and Output nodes. Some filters modify numerical values passing through them; some change data types; some combine or split data, and some just act as intermediate reference points and do not modify data.

[4] A Setup node can provide general information to RTI Server, such as poll classes and association lists; some Setup nodes are involved in controlling associations, and some Setup nodes provide for bulk transfer of input or output data, such as setting up a script or an ICCP transfer set to handle a group of points. In general, a Setup node does not deal with data on a point by point basis. Some Setup nodes have parameters containing a Common Name that refers to a block of network addressing information, which is known as LDIB.

Appendix I: ICCP Configuration Tutorial

In this tutorial, you simulate the dataflow between an RTI Server instance and a simulated SCADA system using ICCP. To simulate the SCADA system, you use a second instance of RTI Server as shown in Figure 53. To generate example data values, you use a custom Python script (Volts.py). Therefore, you will need to create two different configurations by following these steps:

Step 1: Examine an empty configuration in the Node Table.

Step 2: Create a Processor VMD

Step 4: Create an ICCP VMD

Step 5: Create the Script VMD

Step 6: Organize your VMDs to represent the dataflow

Step 7: Create nodes within the ICCP VMD

Step 8: Create nodes within the Processor VMD

Step 9: Create nodes within the Script VMD

Step 10: Create connections between the nodes

Step 11: Specify the properties of RTI Server

Step 12: Load a batch file

Step 13: Start and monitor the ICCP configuration

Step 14: Save and import the ICCP configuration

Step 15: Create a mirror configuration

Figure 53 shows how the ICCP main configuration and the ICCP test configuration communicate.

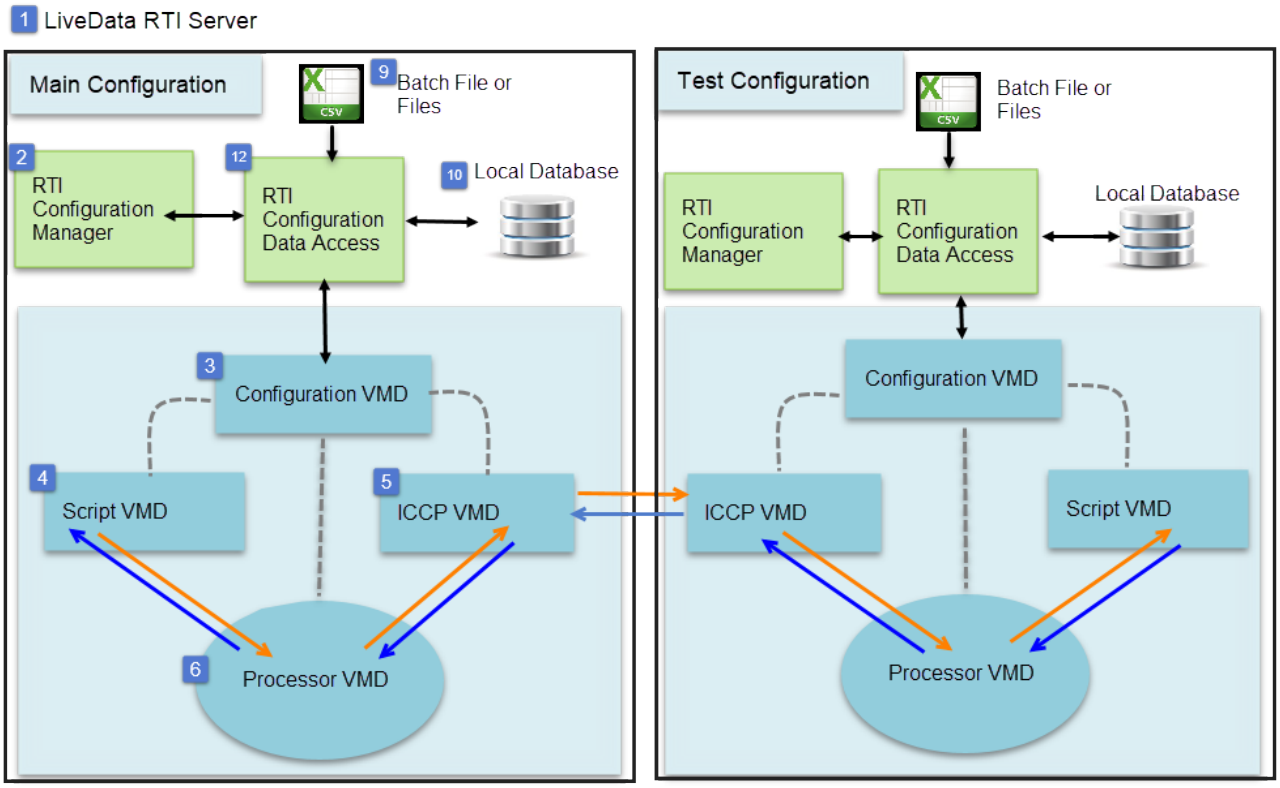
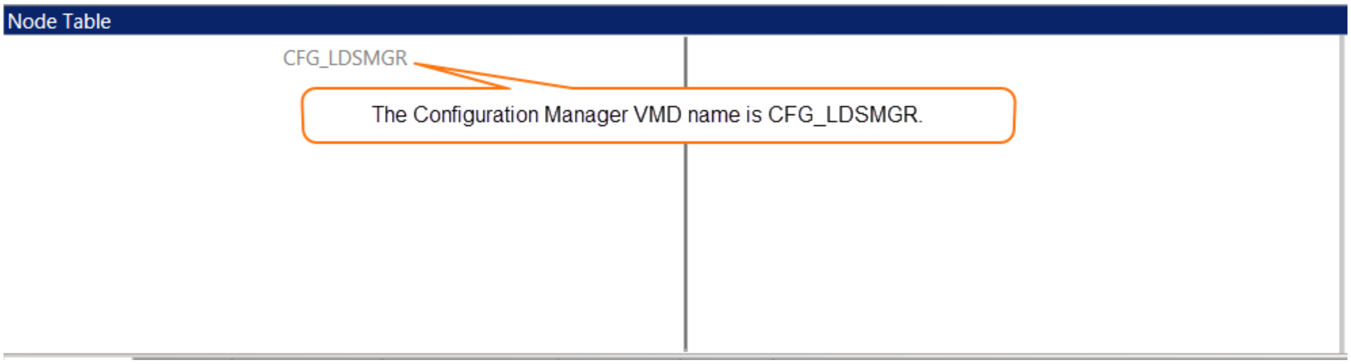


Figure 53: ICCP Main Configuration Data Flow to and from ICCP Test Configuration

Examine an Empty Configuration

Log into RTI Configuration Manager if you are not already logged in.

If you have not imported a configuration database and have not created a VMD, the configuration you see in the Node Table is empty except for the Configuration Manager VMD. Each new configuration starts with a Configuration Manager VMD. The Configuration Manager VMD places the objects you create into RTI Server.



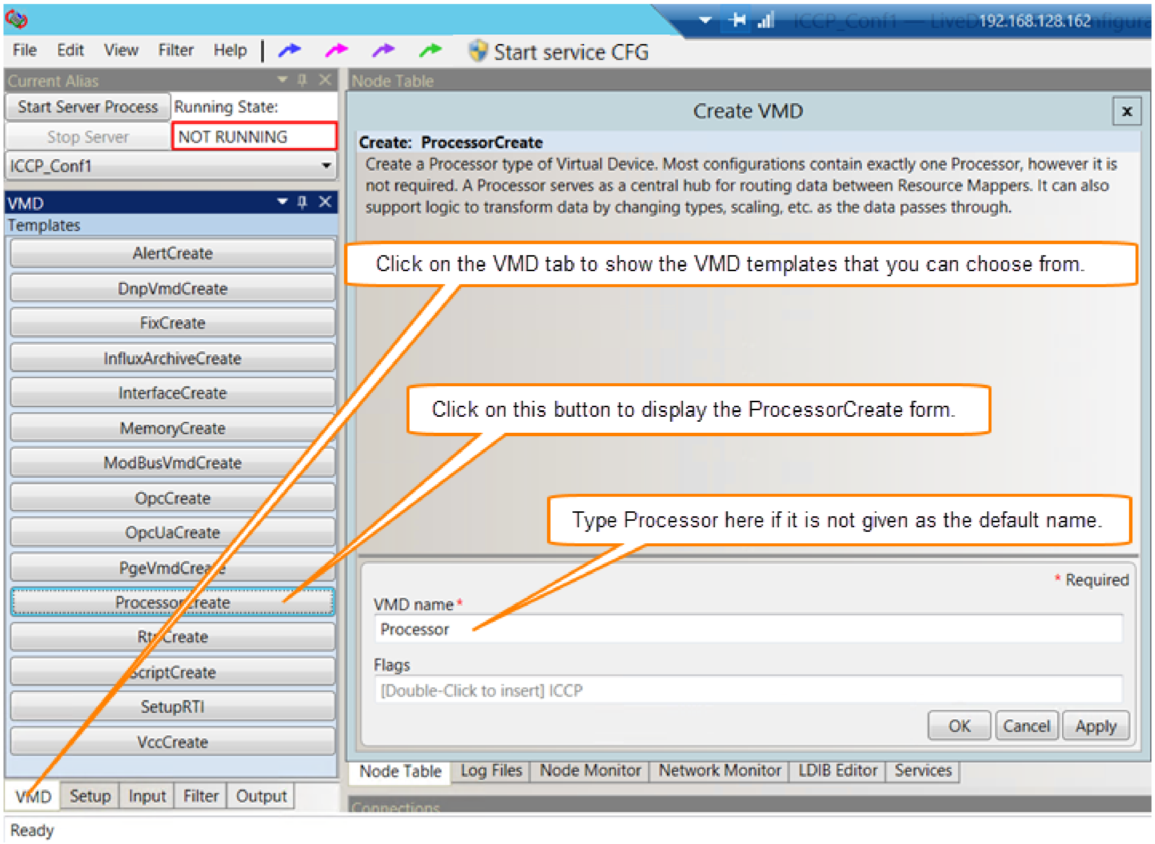
The Configuration Manager VMD name is CFG_LDSMGR.

Now you are going to create three new objects: the Processor VMD, the My_Vcc VMD, and the Script VMD.

Create a Processor VMD

The first VMD that you will create is the processor VMD. Any VMD can communicate to any other VMD by going through the processor VMD.

1. Click on the **ProcessorCreate** template as shown in the next figure.
2. Type in a name for the processor. For the purposes of this tutorial, name it Processor.



Running State: NOT RUNNING

Create: ProcessorCreate

Create a Processor type of Virtual Device. Most configurations contain exactly one Processor, however it is not required. A Processor serves as a central hub for routing data between Resource Mappers. It can also support logic to transform data by changing types, scaling, etc. as the data passes through.

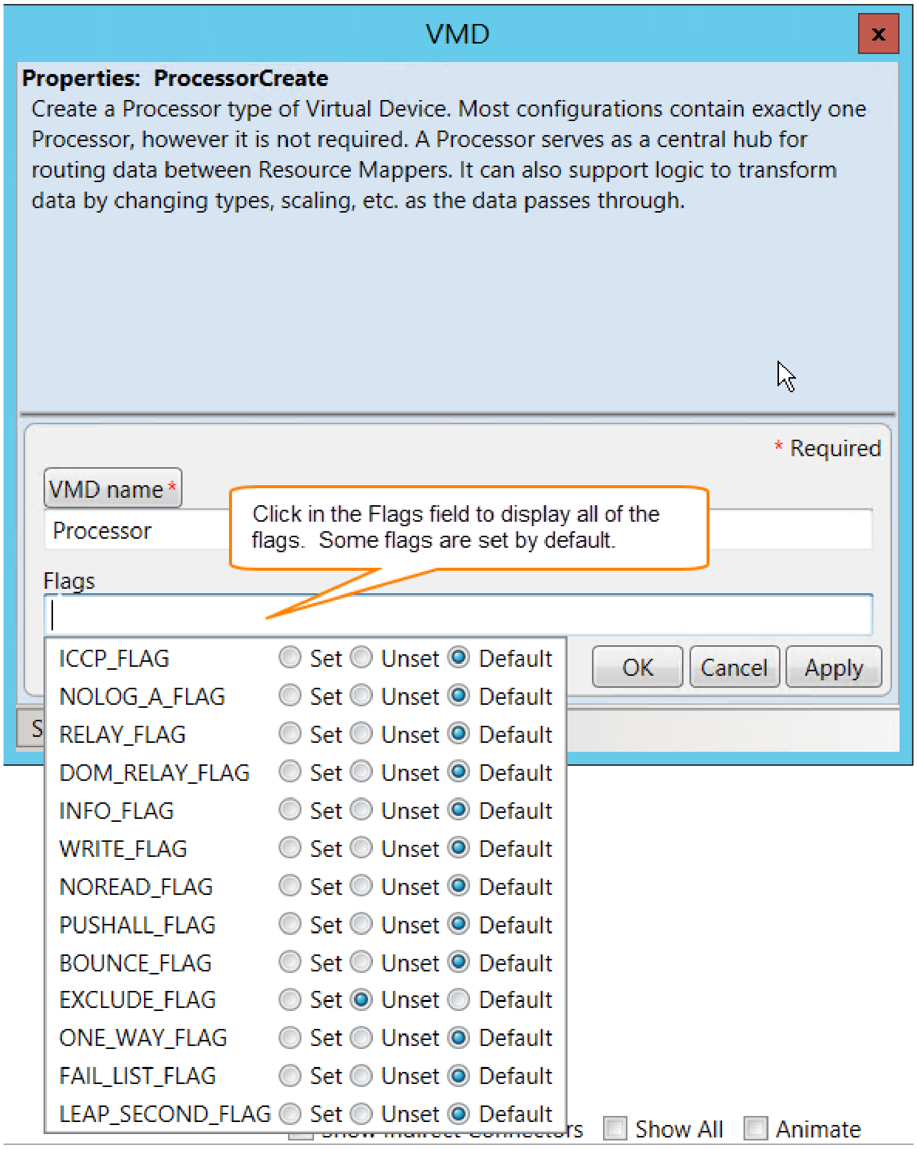
VMD name*: Processor

Flags: [Double-Click to insert] ICCP

Click on the VMD tab to show the VMD templates that you can choose from.

Click on this button to display the ProcessorCreate form.

Type Processor here if it is not given as the default name.



Properties: ProcessorCreate

Create a Processor type of Virtual Device. Most configurations contain exactly one Processor, however it is not required. A Processor serves as a central hub for routing data between Resource Mappers. It can also support logic to transform data by changing types, scaling, etc. as the data passes through.

* Required

VMD name *: Processor

Click in the Flags field to display all of the flags. Some flags are set by default.

Flags:

ICCP_FLAG	○ Set	○ Unset	● Default
NOLOG_A_FLAG	○ Set	○ Unset	● Default
RELAY_FLAG	○ Set	○ Unset	● Default
DOM_RELAY_FLAG	○ Set	○ Unset	● Default
INFO_FLAG	○ Set	○ Unset	● Default
WRITE_FLAG	○ Set	○ Unset	● Default
NOREAD_FLAG	○ Set	○ Unset	● Default
PUSHALL_FLAG	○ Set	○ Unset	● Default
BOUNCE_FLAG	○ Set	○ Unset	● Default
EXCLUDE_FLAG	○ Set	● Unset	○ Default
ONE_WAY_FLAG	○ Set	○ Unset	● Default
FAIL_LIST_FLAG	○ Set	○ Unset	● Default
LEAP_SECOND_FLAG	○ Set	○ Unset	● Default

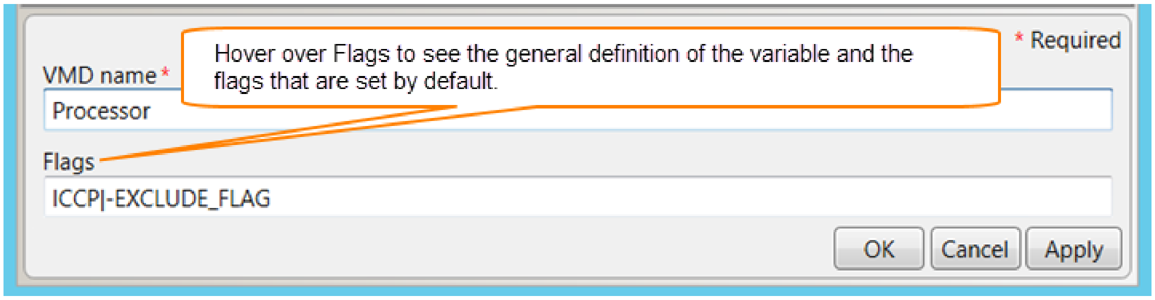
3. Click on the Flags field beneath the VMD name field. A list of flags will be displayed.
4. Unset Exclude_Flag by clicking on the **Unset** radio button.

This argument provides a set of Boolean (on/off) options that enable or disable certain functions in the Processor VMD. The Flags argument is expressed as a series of flag keywords with the vertical bar (I) character between each flag in the list.

The EXCLUDE_FLAG, ICCP_FLAG, PUSHALL_FLAG, and NOREAD_FLAG flags are set by default. To remove a flag, click on the **Unset** button. To set a flag, click on the **Set** button.

Note that you can type a flag name in the text field to set it or precede the flag name with a minus sign to unset it as shown in the following example.

5. Click on the main form and then click the **OK** button to create the Processor VMD.



Hover over Flags to see the general definition of the variable and the flags that are set by default.

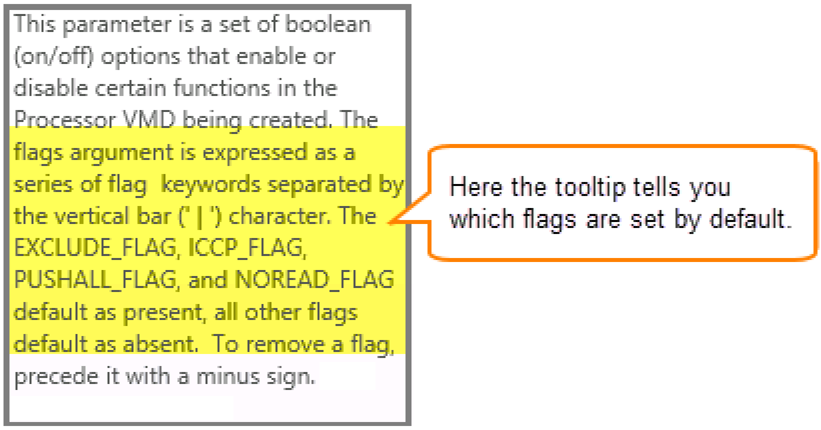
* Required

VMD name*: Processor

Flags: ICCP|-EXCLUDE_FLAG

In this example, the EXCLUDE_FLAG is unset, and the ICCP_FLAG is set. The EXCLUDE_FLAG is unset because if it is set, the Processor VMD cannot receive data from incoming connections.

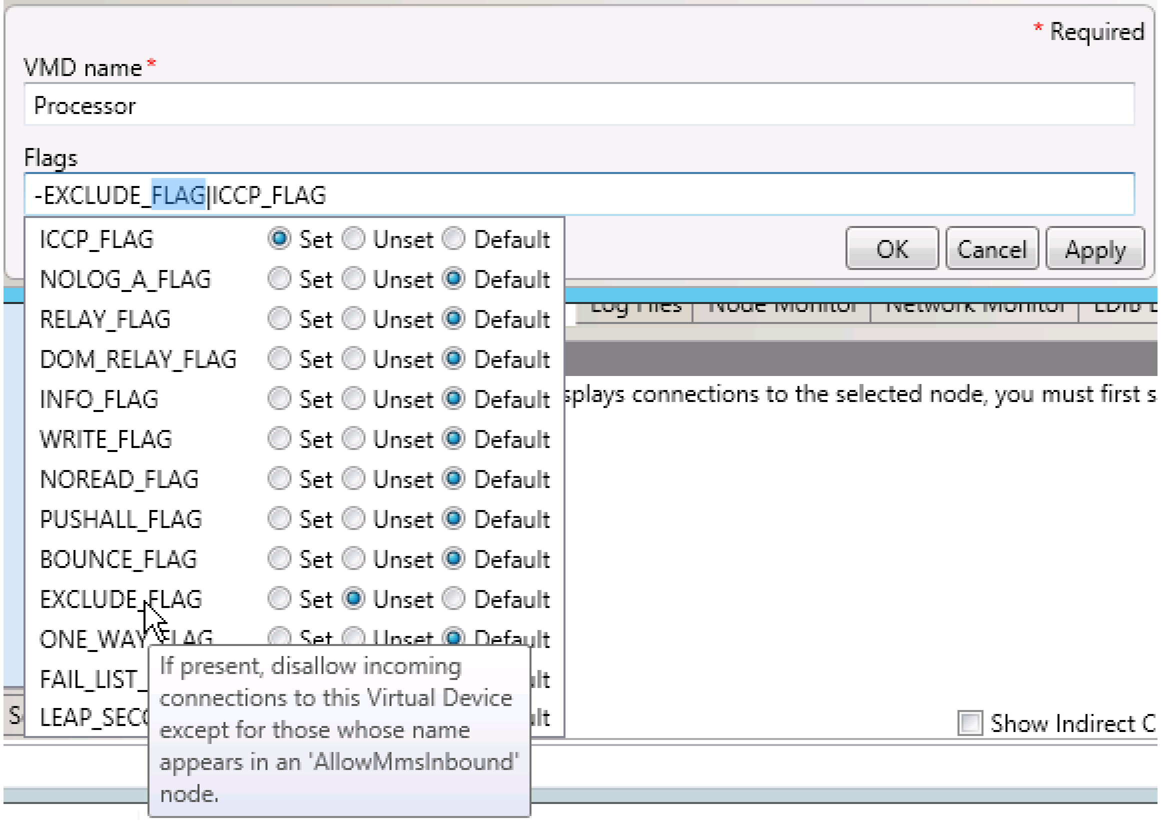
To find information about the flags, hover over the label Flags in the form and a general description is displayed.



This parameter is a set of boolean (on/off) options that enable or disable certain functions in the Processor VMD being created. The flags argument is expressed as a series of flag keywords separated by the vertical bar (' | ') character. The EXCLUDE_FLAG, ICCP_FLAG, PUSHALL_FLAG, and NOREAD_FLAG default as present, all other flags default as absent. To remove a flag, precede it with a minus sign.

Here the tooltip tells you which flags are set by default.

In addition, you can hover over a particular flag within the form itself to display help on that particular flag as shown in the following example.



* Required

VMD name*: Processor

Flags: -EXCLUDE_FLAG|ICCP_FLAG

ICCP_FLAG	● Set	○ Unset	○ Default
NOLOG_A_FLAG	○ Set	○ Unset	● Default
RELAY_FLAG	○ Set	○ Unset	● Default
DOM_RELAY_FLAG	○ Set	○ Unset	● Default
INFO_FLAG	○ Set	○ Unset	● Default
WRITE_FLAG	○ Set	○ Unset	● Default
NOREAD_FLAG	○ Set	○ Unset	● Default
PUSHALL_FLAG	○ Set	○ Unset	● Default
BOUNCE_FLAG	○ Set	○ Unset	● Default
EXCLUDE_FLAG	○ Set	● Unset	○ Default
ONE_WAY_FLAG	○ Set	○ Unset	● Default
FAIL_LIST_			lt
LEAP_SEC			lt

If present, disallow incoming connections to this Virtual Device except for those whose name appears in an 'AllowMmsInbound' node.

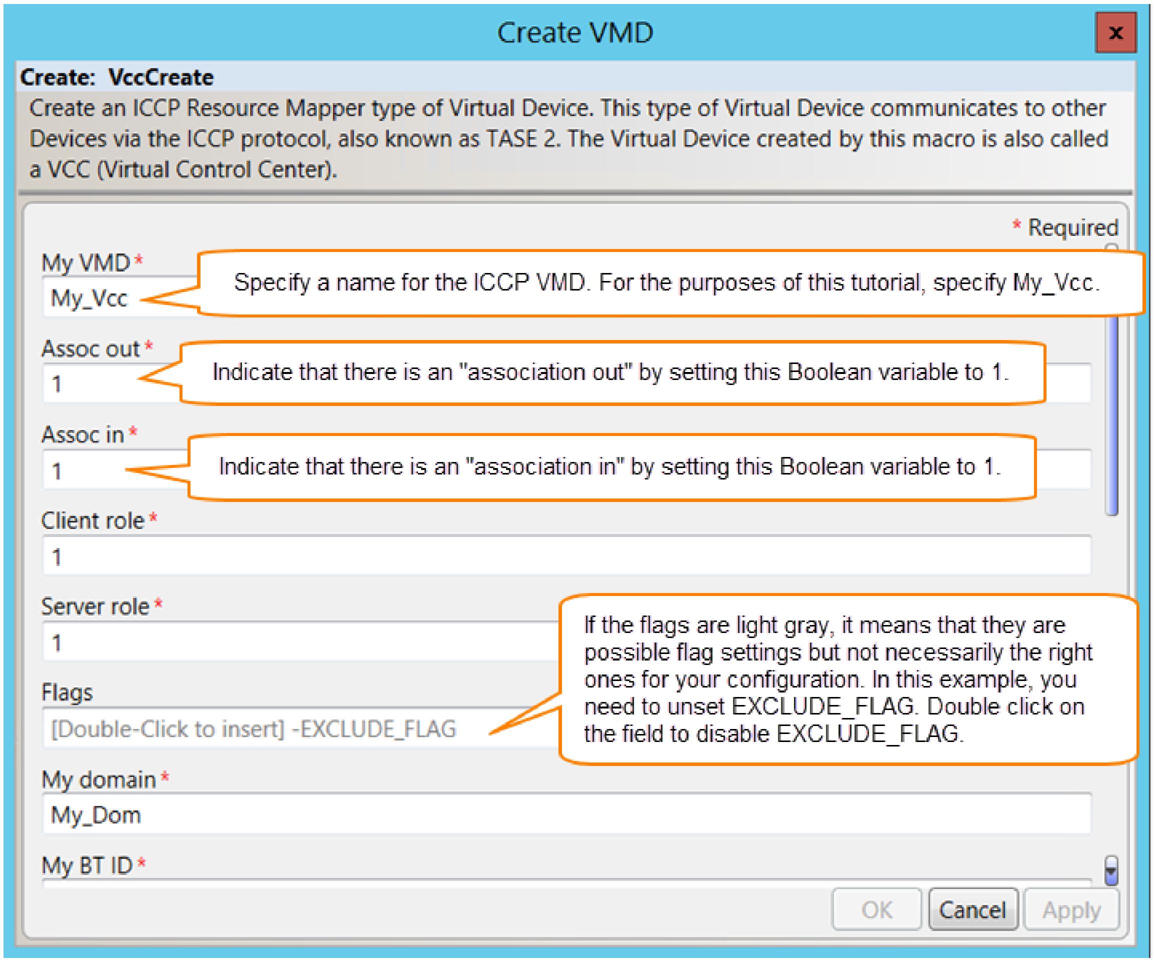
6. Once you have created the Processor VMD, you can set or unset flags in the same way, except that you apply your changes with the **Apply** button instead of the **OK** button.

You can hover over any of the labels in the Template forms to display tooltips on the label of interest.

Create an ICCP VMD

In order to create an ICCP VMD:

1. Select the VccCreate template from the Template menu.



Create VMD

Create: VccCreate

Create an ICCP Resource Mapper type of Virtual Device. This type of Virtual Device communicates to other Devices via the ICCP protocol, also known as TASE 2. The Virtual Device created by this macro is also called a VCC (Virtual Control Center).

* Required

My VMD *

My_Vcc

Specify a name for the ICCP VMD. For the purposes of this tutorial, specify My_Vcc.

Assoc out *

1

Indicate that there is an "association out" by setting this Boolean variable to 1.

Assoc in *

1

Indicate that there is an "association in" by setting this Boolean variable to 1.

Client role *

1

Server role *

1

Flags

[Double-Click to insert] -EXCLUDE_FLAG

If the flags are light gray, it means that they are possible flag settings but not necessarily the right ones for your configuration. In this example, you need to unset EXCLUDE_FLAG. Double click on the field to disable EXCLUDE_FLAG.

My domain *

My_Dom

My BT ID *

2. Specify the name of the ICCP VMD, which is known as a VCC (short for Virtual Control Center), beneath the label My VMD. For the purpose of this tutorial, specify My_Vcc.

3. Set both the Assoc In and the Assoc Out fields to 1, allowing for outgoing and incoming associations.

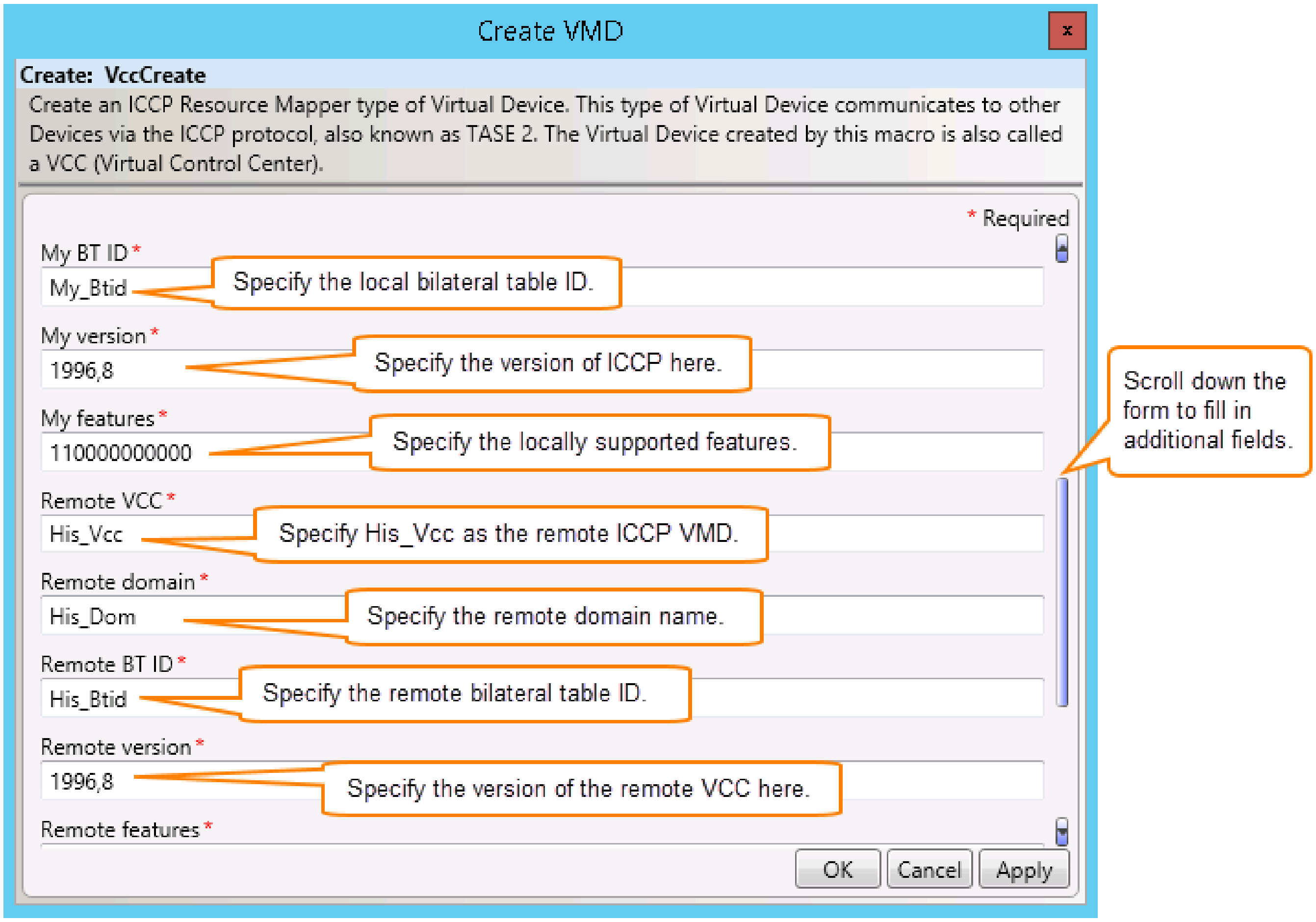
4. Set Client role to 1.

5. Set Server role to 1. In this example, each VCC will serve both roles; however, you can limit a VCC to serve in only one role.

6. If you need to set a flag, click on the Flags field as you did to set and unset the flags for the Processor VMD. If you would like to accept the gray flag settings, double click on the field. (To unset a flag, type the minus sign in front of the flag name, as in -EXCLUDE_FLAG.) In this, case, EXCLUDE_FLAG is unset for you because it already has a minus sign in front of it. All you need to do is to double click on the field.

7. Specify the domain of the server in the My domain field. By default, this field is set to My_Dom. Accept the default setting.

8. Scroll down the form.



Create: VccCreate

Create an ICCP Resource Mapper type of Virtual Device. This type of Virtual Device communicates to other Devices via the ICCP protocol, also known as TASE 2. The Virtual Device created by this macro is also called a VCC (Virtual Control Center).

* Required

My BT ID *: My_Btid

Specify the local bilateral table ID.

My version *: 1996,8

Specify the version of ICCP here.

My features *: 110000000000

Specify the locally supported features.

Remote VCC *: His_Vcc

Specify His_Vcc as the remote ICCP VMD.

Remote domain *: His_Dom

Specify the remote domain name.

Remote BT ID *: His_Btid

Specify the remote bilateral table ID.

Remote version *: 1996,8

Specify the version of the remote VCC here.

Remote features *

Scroll down the form to fill in additional fields.

9. Use My_Btid in the My BT ID field. This field specifies the local bilateral table ID.

10. Specify the version of the local ICCP application. By default, RTI Configuration Manager sets this value to the correct version, such as 1996,8.

11. Specify the locally supported features in a Boolean bit string of 12 Boolean values (1 or 0). During Association establishment, the client accesses the ICCP standard object named Supported_Features. This object returns a bit string that specifies the ICCP Conformance Blocks supported in the server. In this case, the server supports the first two blocks.

12. Specify the name of the remote ICCP VMD. The name given to the remote VCC in this example is His_Vcc.

13. Specify the remote domain name as His_Dom.

14. Specify the remote bilateral table ID as His_Btid.

15. Specify the version of the remote ICCP version, which must be the same as the local version.

16. Continue to scroll down to the end of the form.

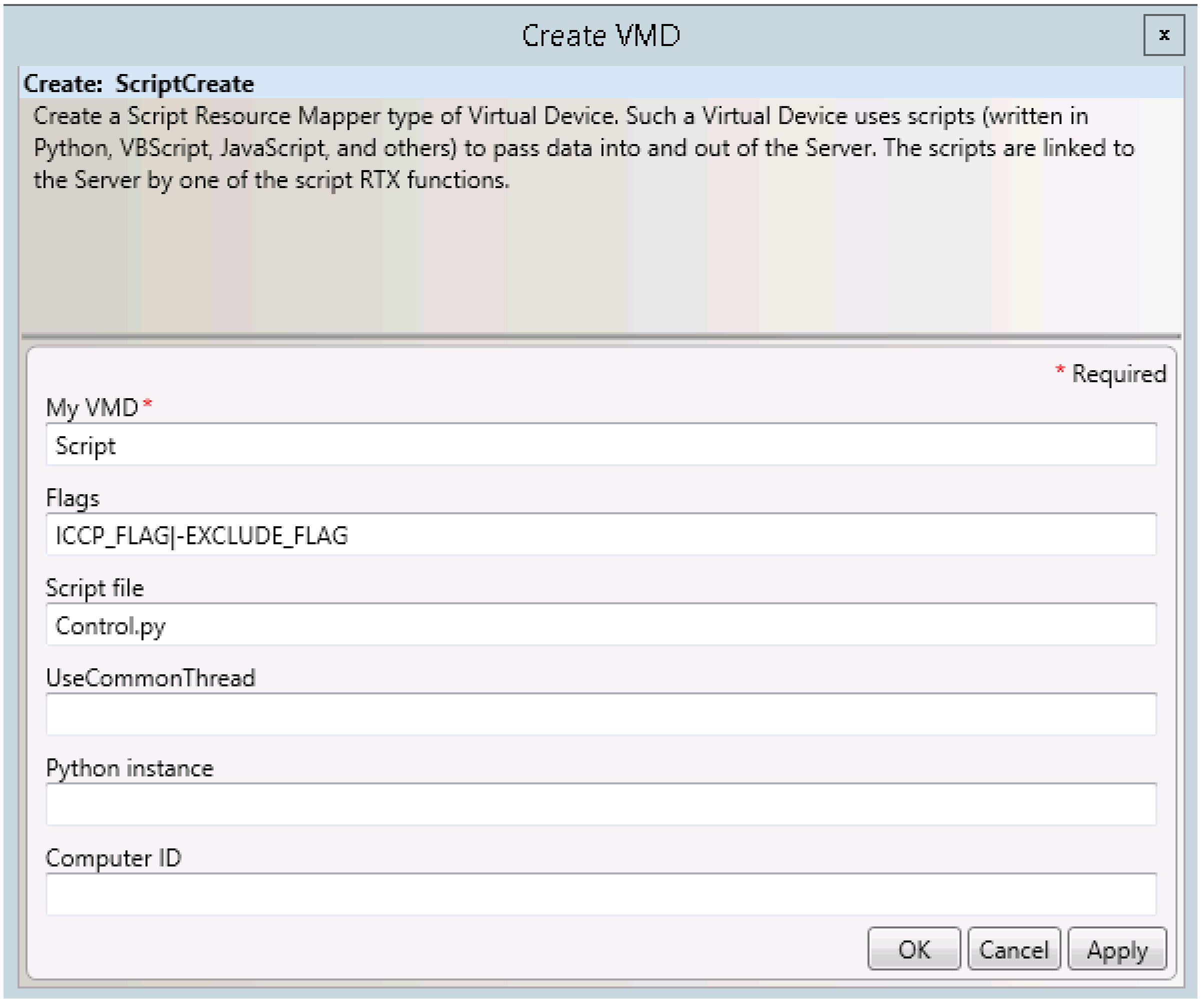
17. Specify the remote features. These must duplicate those that are supported locally, 110000000000.

18. Leave Delay base, Value poll time, and Value timeout blank.

19. Click **OK** to create this VMD.

Create the Script VMD

The Script VMD feeds values into the Processor nodes, which are then read into the My_Vcc nodes. Figure 54 shows the Create VMD form filled in with the values required for the purpose of this ICCP configuration. The steps you need to take to create the Script VMD follow the figure.



Create VMD

Create: ScriptCreate

Create a Script Resource Mapper type of Virtual Device. Such a Virtual Device uses scripts (written in Python, VBScript, JavaScript, and others) to pass data into and out of the Server. The scripts are linked to the Server by one of the script RTX functions.

* Required

My VMD *
Script

Flags
ICCP_FLAG|-EXCLUDE_FLAG

Script file
Control.py

UseCommonThread

Python instance

Computer ID

Figure 54: Create a Script VMD

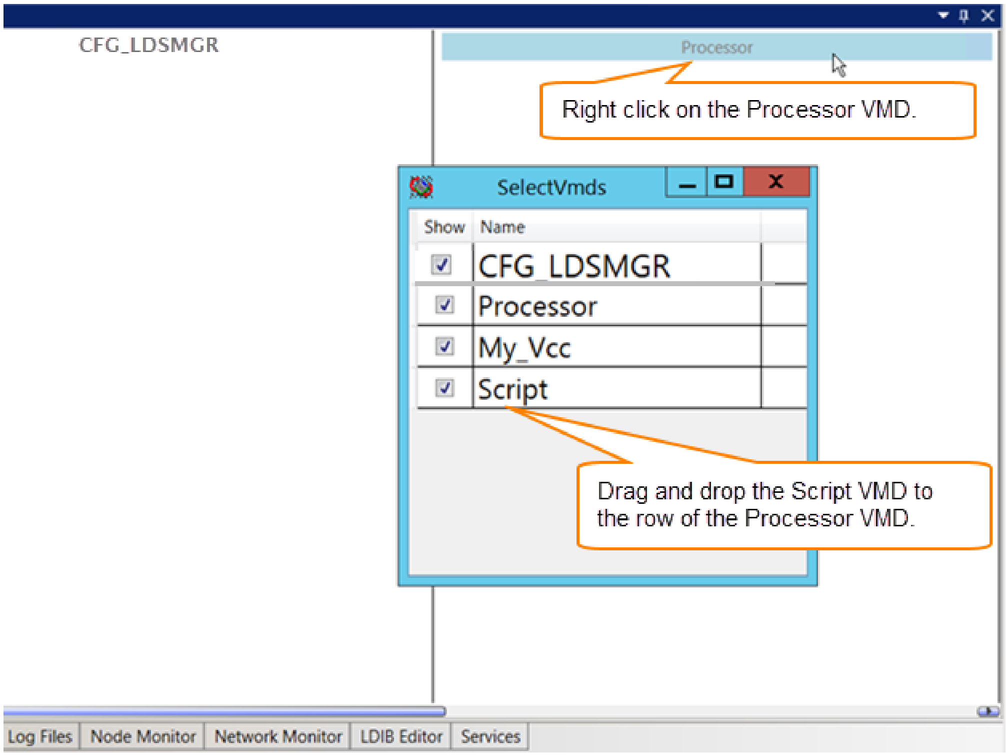
1. Select the VMD tab under the Templates panel.
2. Select the ScriptCreate template, which will display the Create VMD form as shown in Figure 54.
3. Type the name Script in the My VMD field. First, you may need to delete the name "My_Vcc" from the field.
4. Specify the flag settings ICCP_FLAG | -EXCLUDE_FLAG.
5. Specify the Script file to be Control.py.
6. Click **OK**.

Organize Your VMDs to Represent the Dataflow

Before going on to create the nodes and connections beneath each VMD, you should understand the flow of the data from the Script VMD to a Processor VMD, where you can read the script-generated data, and finally to a local VCC that is designed to pass this data to a remote VCC.

To do this:

1. Right click on the Processor VMD as shown in the next figure.



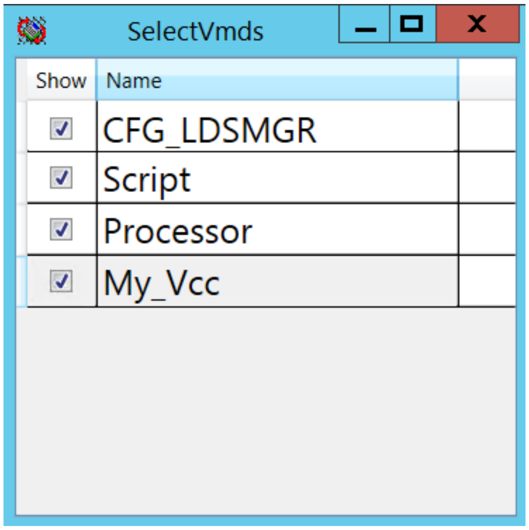
Right click on the Processor VMD.

Show	Name
☑	CFG_LDSMGR
☑	Processor
☑	My_Vcc
☑	Script

Drag and drop the Script VMD to the row of the Processor VMD.

2. Drag and drop the Script VMD to where the Processor VMD is.
3. This will change the Node Table to show the VMDs in the following order:

Script to **Processor** to **My_Vcc**



Show	Name
☑	CFG_LDSMGR
☑	Script
☑	Processor
☑	My_Vcc

To hide a VMD, click on the checkbox beside the VMD you would like to hide.

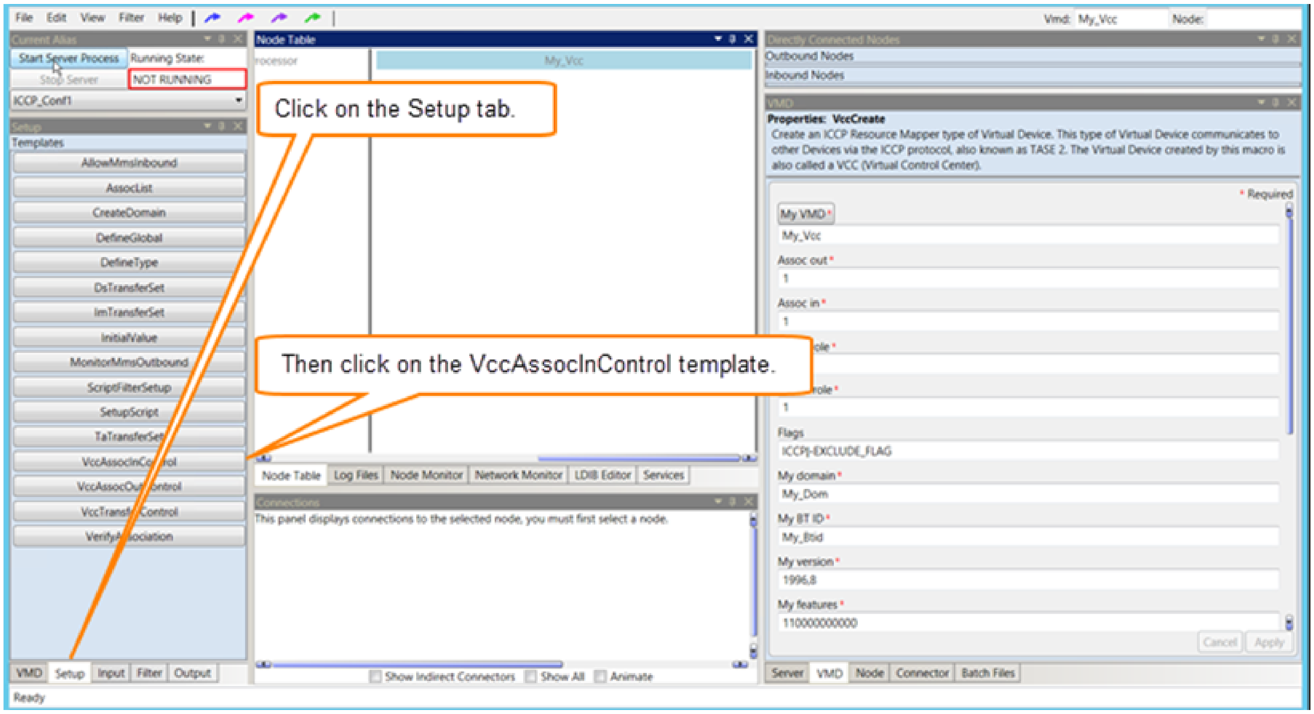
To return RTI Configuration Manager to display the hidden VMD, click on the checkbox again.

Create Nodes within the ICCP VMD

In this section, you will create the nodes that ICCP requires to set up an association between this VCC and a remote VCC.

Create a Setup Node within My_Vcc for Inbound Association Control

The Inbound Association Control node enables and disables the inbound ICCP association to this VCC, and reports the association status.



Click on the Setup tab.

Then click on the VccAssocInControl template.

Running State: NOT RUNNING

This panel displays connections to the selected node, you must first select a node.

Properties: VccCreate

Create an ICCP Resource Mapper type of Virtual Device. This type of Virtual Device communicates to other Devices via the ICCP protocol, also known as TASE 2. The Virtual Device created by this macro is also called a VCC (Virtual Control Center).

Vmd: My_Vcc

My VMD *: My_Vcc

Assoc out *: 1

Assoc in *: 1

Flags: ICCPJ-EXCLUDE_FLAG

My domain *: My_Dom

My BT ID *: My_Btid

My version *: 1996,8

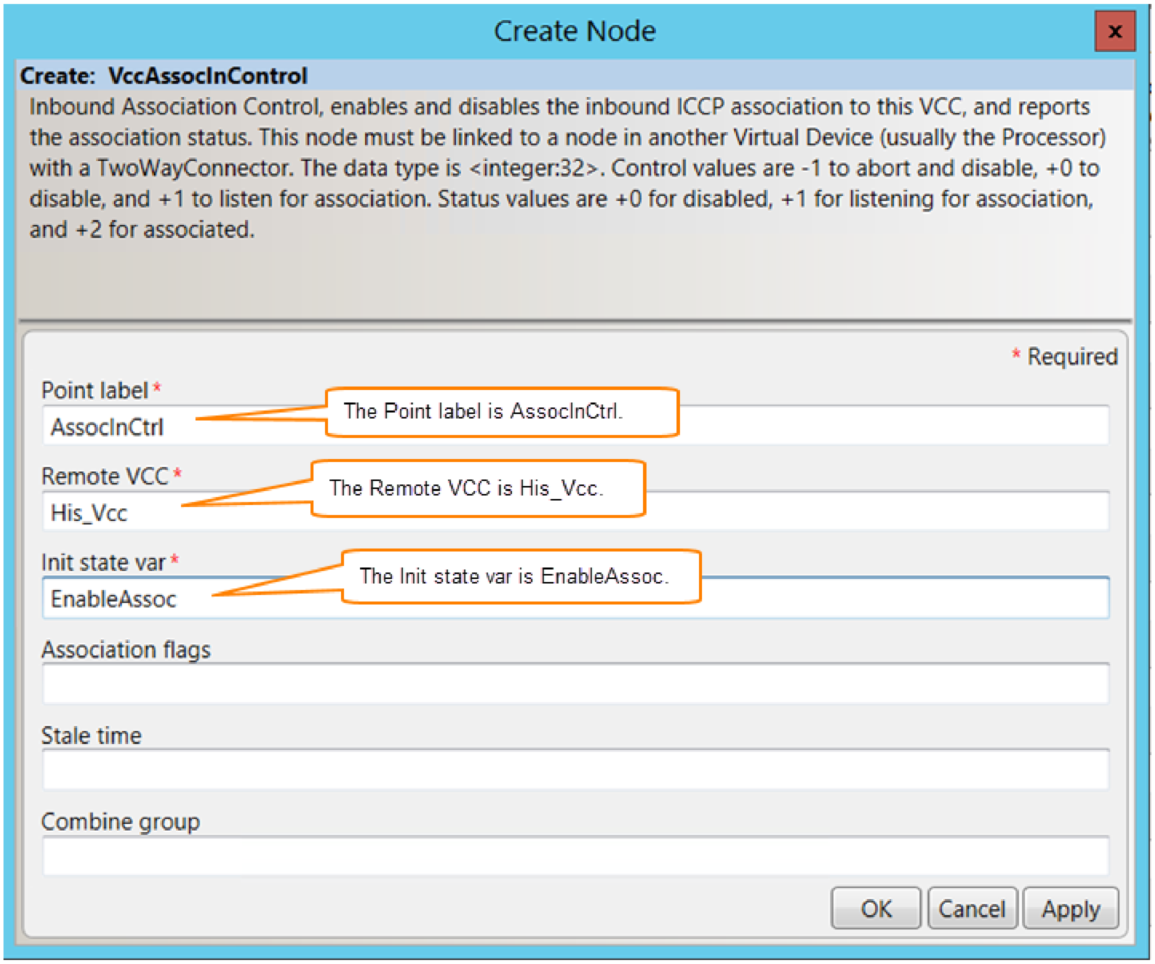
My features *: 110000000000

To create a Setup node for Inbound Association Control:

1. Click on My_Vcc in the Node Table. A transparent blue bar will be displayed over the VMD name to indicate that this device has been selected.

2. Click on the Setup tab at the bottom of the Templates panel.

3. Click on the VccAssocInControl template to display the Create Node form. RTI Configuration Manager correctly fills in the remote Vcc as His_Vcc. RTI Configuration Manageralready has this information because the remote VCC is already defined within My_Vcc.



Create Node

Create: VccAssocInControl

Inbound Association Control, enables and disables the inbound ICCP association to this VCC, and reports the association status. This node must be linked to a node in another Virtual Device (usually the Processor) with a TwoWayConnector. The data type is <integer:32>. Control values are -1 to abort and disable, +0 to disable, and +1 to listen for association. Status values are +0 for disabled, +1 for listening for association, and +2 for associated.

* Required

Point label *: AssocInCtrl

The Point label is AssocInCtrl.

Remote VCC *: His_Vcc

The Remote VCC is His_Vcc.

Init state var *: EnableAssoc

The Init state var is EnableAssoc.

Association flags:

Stale time:

Combine group:

1. Read the description of VccAssocInControl. The last sentence of the description indicates the purpose of the control values. Of specific note, +1 instructs the node to listen for an association from a remote node. Therefore, you will initialize the state variable to +1 in Step 3.

2. Specify the Point label as AssocInCtrl for the purposes of this tutorial.

3. Specify the Init state var as **EnableAssoc**. EnableAssoc is a constant that is set to +1 later in this section. This setting will cause this VCC to listen for an association as mentioned above.

4. Leave Association flags, Stale time, and Combine group blank.

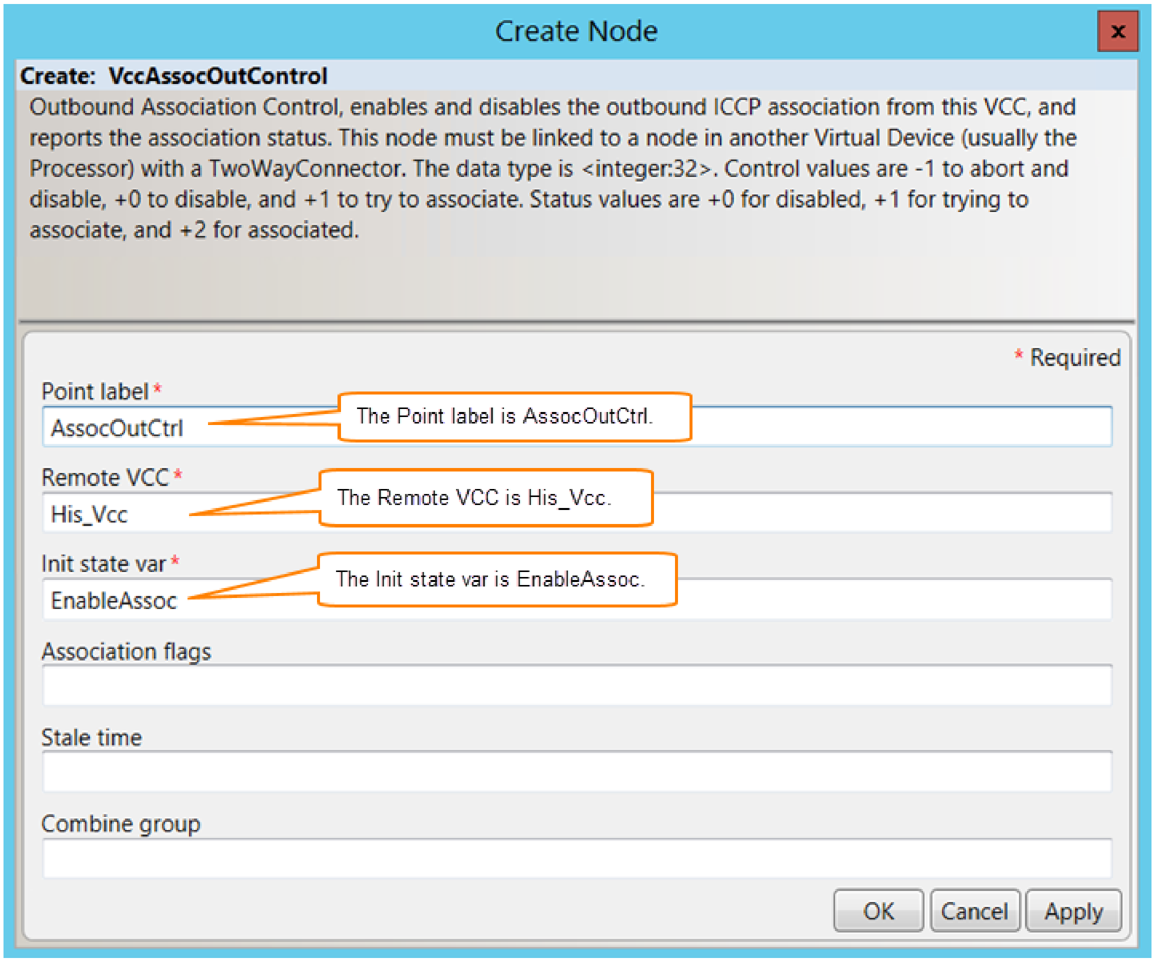
5. Click on the **OK** button on the bottom of the form to create the node.

Create a Setup Node within My_Vcc for Outbound Association Control

The Outbound Association Control node enables and disables the outbound ICCP association from this VCC, and reports the association status.

1. Click on My_Vcc in the Node Table.

2. From the Setup tab, select the VccAssocOutControl template to bring up the Create Node form.



Create: VccAssocOutControl

Outbound Association Control, enables and disables the outbound ICCP association from this VCC, and reports the association status. This node must be linked to a node in another Virtual Device (usually the Processor) with a TwoWayConnector. The data type is <integer:32>. Control values are -1 to abort and disable, +0 to disable, and +1 to try to associate. Status values are +0 for disabled, +1 for trying to associate, and +2 for associated.

* Required

Point label *: AssocOutCtrl

Remote VCC *: His_Vcc

Init state var *: EnableAssoc

Association flags:

Stale time:

Combine group:

The Point label is AssocOutCtrl.

The Remote VCC is His_Vcc.

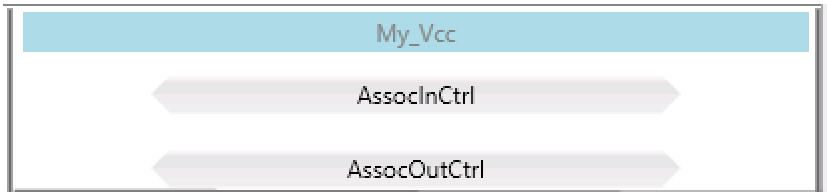
The Init state var is EnableAssoc.

1. Specify the Point label as AssocOutCtrl.

2. Specify the additional fields as you did in the VccAssocInControl Create Node form. Similar to AssocInCtrl, the Init state var is set to the value of the constant EnableAssoc. In this case, the value is +1 causes the VCC to attempt to make an association with the remote VCC, His_Vcc,

3. Click **OK** to finish creating the node.

Now the Node Table will show the two nodes you created within My_Vcc.



My_Vcc

AssocInCtrl

AssocOutCtrl

Create an Input Node that Is a Constant

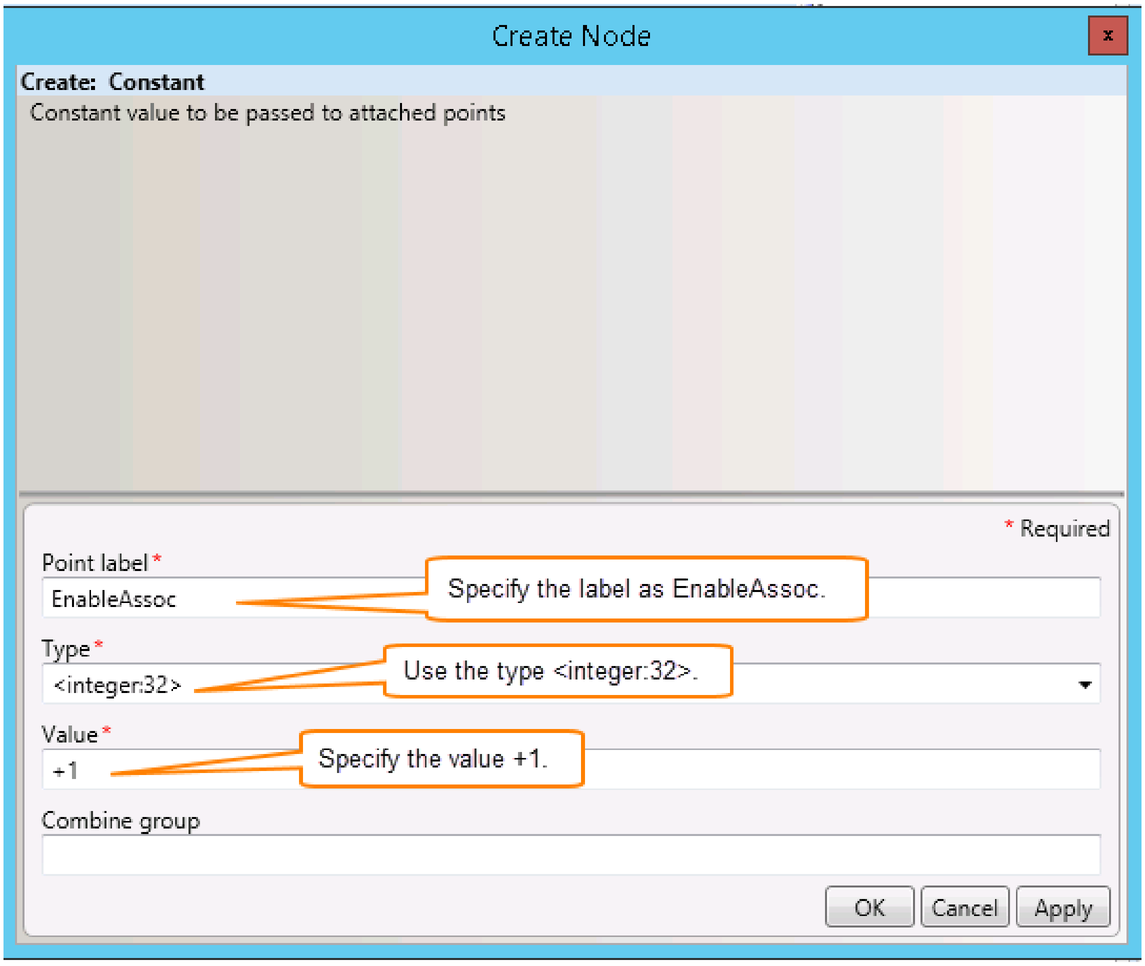
This constant is named **EnableAssoc**, and it initializes the value of Init state var in **AssocInCtrl** and **AssocOutCtrl.**

To create this constant:

1. Click on My_Vcc if it is not already selected.

2. Click on the Input tab at the bottom of the Templates panel.

3. Click on the Constant template to display the Create Node form, which RTI Configuration Manager will partially fill in for you.



Create: Constant

Constant value to be passed to attached points

* Required

Point label*
EnableAssoc

Specify the label as EnableAssoc.

Type*
<integer:32>

Use the type <integer:32>.

Value*
+1

Specify the value +1.

Combine group

1. Specify the Point label as **EnableAssoc**.
2. Specify the Type as <integer:32>.
3. Specify the Value as +1.
4. Click on the **OK** button to create this constant.

Create another Constant within My_Vcc

The constant is named **EnableXfer**. RTI Server passes this constant value to XferCtrl. The XferCtrl node enables or disables transfer sets according to the value in EnableXfer.

To create this constant:

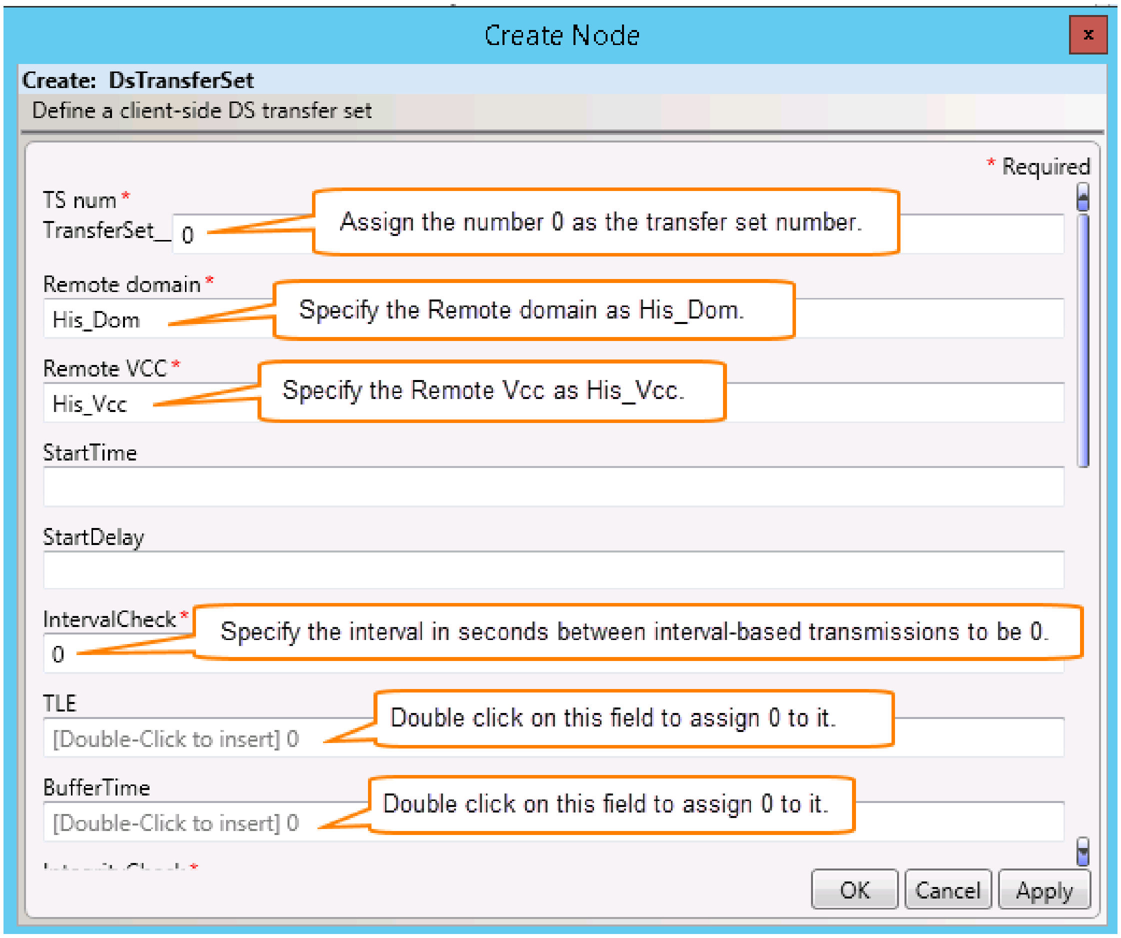
1. Click on the Input tab at the bottom of the Templates panel.
2. Click on the Constant template.
3. Specify the Point label as EnableXfer.
4. Specify the type as <boolean>(<pack>16).
5. Specify the value as [T,T,F,F,F,F,F,F,F,F,F,F,F,F,F,F]. The first two elements of this array are set to true, enabling the first two transfer sets. The remaining 14 elements are set to false, disabling the other transfer sets.

Create Another Setup Node within My_Vcc for the DS Transfer Set

The DS transfer set defines the client-side data set.

To define the DS transfer set:

1. Select My_Vcc.
2. Click on the Setup tab at the bottom of the Templates panel.
3. Click on the DsTransferSet template to display the Create Node form.



Create Node

Create: DsTransferSet

Define a client-side DS transfer set

* Required

TS num *
TransferSet_ 0

Assign the number 0 as the transfer set number.

Remote domain *
His_Dom

Specify the Remote domain as His_Dom.

Remote VCC *
His_Vcc

Specify the Remote Vcc as His_Vcc.

StartTime

StartDelay

IntervalCheck *
0

Specify the interval in seconds between interval-based transmissions to be 0.

TLE
[Double-Click to insert] 0

Double click on this field to assign 0 to it.

BufferTime
[Double-Click to insert] 0

Double click on this field to assign 0 to it.

4. Assign the value 0 to the TS num field to create the unique name TransferSet_0 that will identify this particular transfer set.

5. Set Remote domain to His_Dom.

6. Set Remote_Vcc to His_Vcc.

7. Leave StartTime and StartDelay blank.

8. Set IntervalCheck to 0 to ensure that RTI Server does not wait between transmissions.

9. Double clicking on gray fields is a quick way to accept the given value if the given value is correct. In this form, ensure that the following fields are set to 0:

- TLE: Zero sets the time limit for execution in seconds to 0.
- BufferTime: Zero sets the buffer time to 0. Buffer time is set in seconds.
- IntegrityCheck: Zero indicates the period in seconds between integrity-based transmissions.
- IntervalTimeOut : Zero indicates that interval-based transmissions are not enabled.
- IntegrityTimeOut: Zero disables integrity-based transmissions. This feature is used to occasionally obtain the state of all points in the transfer set when operating in Report By Exception mode. RTI Server sends only the points that have changed in this mode.

10. Specify Object Change to be 1 in order to transmit data whenever there is a change.

11. Specify OperatorRequest to be 0. Zero disables operator-requested transmissions.

12. Specify RBE to be 1 to enable report by exception.

13. Leave IntervalDelay and AllChangesReported blank.

14. Specify Critical to be 0. Zero indicates that critical acknowledgment is not required.

15. Specify CircumventSiemensBug to be 0. Zero indicates that there is no need to circumvent a Siemens bug since you are not working with Siemens.
16. Specify 1 in the Do read field to indicate that the node should do an initial read of the data set.
17. Leave Combine group blank.
18. Click on the **OK** button to create the node.

Create the XferCtrl Setup Node

This node enables and disables the initial state of transfer sets and reports transfer set status.

1. Select the My_Vcc VMD.
2. Select the Setup tab in the Templates panel.
3. Select the VccTransferControl template.
4. Set the Point label to XferCtrl in the Create Node form.
5. Set the Type field to <boolean>(<pack>16).
6. Set the Init state var field to EnableXfer to enable the first two transfer sets.

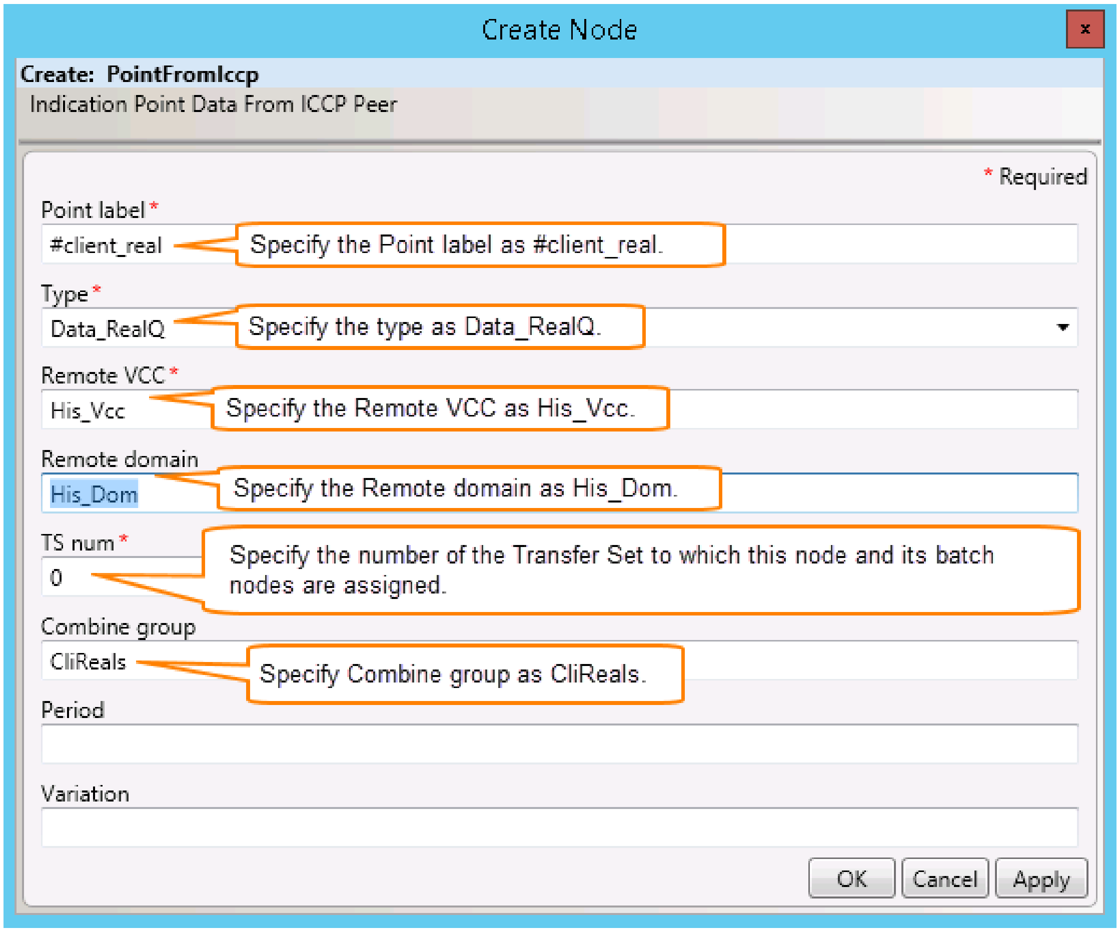
Create a Prototype Input Node within My_Vcc

A prototype node provides a template from which other nodes can be created. They do not represent devices, nor do they contain data since they serve only as templates to create other nodes that will actually represent points. You are defining this prototype node as a template for Input nodes that are able to receive data from the remote node His_Vcc.

1. Click on My_Vcc to select it.
2. Click on the Input tab at the bottom of the Templates panel.
3. Select the PointFromIccp template to display the Create Node form. Prototype nodes begin with a pound sign (#).

Figure 55 provides all of the information that you will need to fill in this Create Node form. Note that the data type is Data_RealQ. RTI Server interprets a variable or point with this data type as a real number with a quality byte. The quality of the transmission is denoted by the specific bit or bits that are set in the byte.

For example, the real number might be 123.664 and the quality byte might appear as '00000010'b.



Create Node

Create: PointFromIccp
Indication Point Data From ICCP Peer

* Required

Point label *: #client_real
Specify the Point label as #client_real.

Type *: Data_RealQ
Specify the type as Data_RealQ.

Remote VCC *: His_Vcc
Specify the Remote VCC as His_Vcc.

Remote domain: His_Dom
Specify the Remote domain as His_Dom.

TS num *: 0
Specify the number of the Transfer Set to which this node and its batch nodes are assigned.

Combine group: CliReals
Specify Combine group as CliReals.

Period:

Variation:

Figure 55: PointFromIccp Form

4. Note that the TS num value correlates to the transfer set number that is appended to the string TransferSet to form the transfer set name TransferSet_0, which you defined in the node DsTransferSet.

5. Click **OK** to create the node.

The prototype nodes are never passed to RTI Server; they are stored in RTI Configuration Manager. Only the points that are generated from the prototype nodes are passed to RTI Server.

Create another Prototype Input Node within My_Vcc

Create another prototype Input Node as you did for #client_real, except that this node is named #client_state and has the Type Data_StateQ. A variable or point with this data type is a discrete 2-bit value plus 6 ICCP Quality bits. The quality of the transmission is denoted by the specific bit or bits that are set in the six quality bits.

An example of a Data_StateQ value is '10000000'.

State is shown in the first pair.

00 means between states..

01 means Off.

10 means On.

11 means Invalid.

Validity is shown in the second pair of bits.

00 means Good or Valid.

01 means Held.

10 means Suspect.

11 means Bad or Invalid.

Current Source is shown in the third pair of bits.

00 means Telemetered.

01 means Calculated.

10 means Entered.

11 means Estimated.

Normal or Abnormal Value is shown in the next single bit.

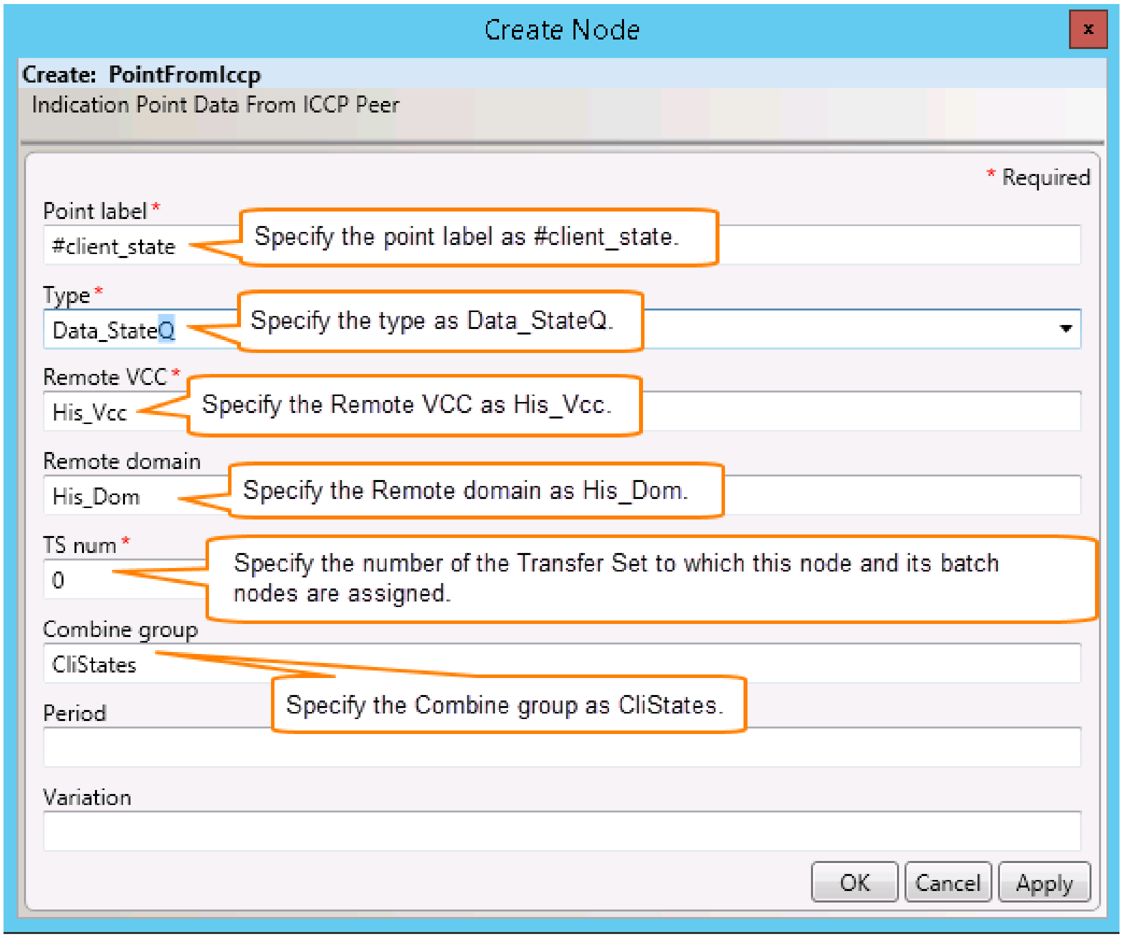
0 means Normal.

1 means Abnormal.

Valid or Invalid Timestamp is shown in the last single bit.

0 means Valid.

1 means Invalid.



Create Node

Create: PointFromIccp

Indication Point Data From ICCP Peer

* Required

Point label *: #client_state

Type *: Data_StateQ

Remote VCC *: His_Vcc

Remote domain: His_Dom

TS num *: 0

Combine group: CliStates

Period:

Variation:

Specify the point label as #client_state.

Specify the type as Data_StateQ.

Specify the Remote VCC as His_Vcc.

Specify the Remote domain as His_Dom.

Specify the number of the Transfer Set to which this node and its batch nodes are assigned.

Specify the Combine group as CliStates.

6. Note again that the TS num value correlates to the transfer set defined in the DsTransferSet node.
7. Specify CliStates for the Combine group.
8. Click **OK** to create the node.
9. You can monitor the value of this Data_StateQ value in the Node Monitor later in this chapter.

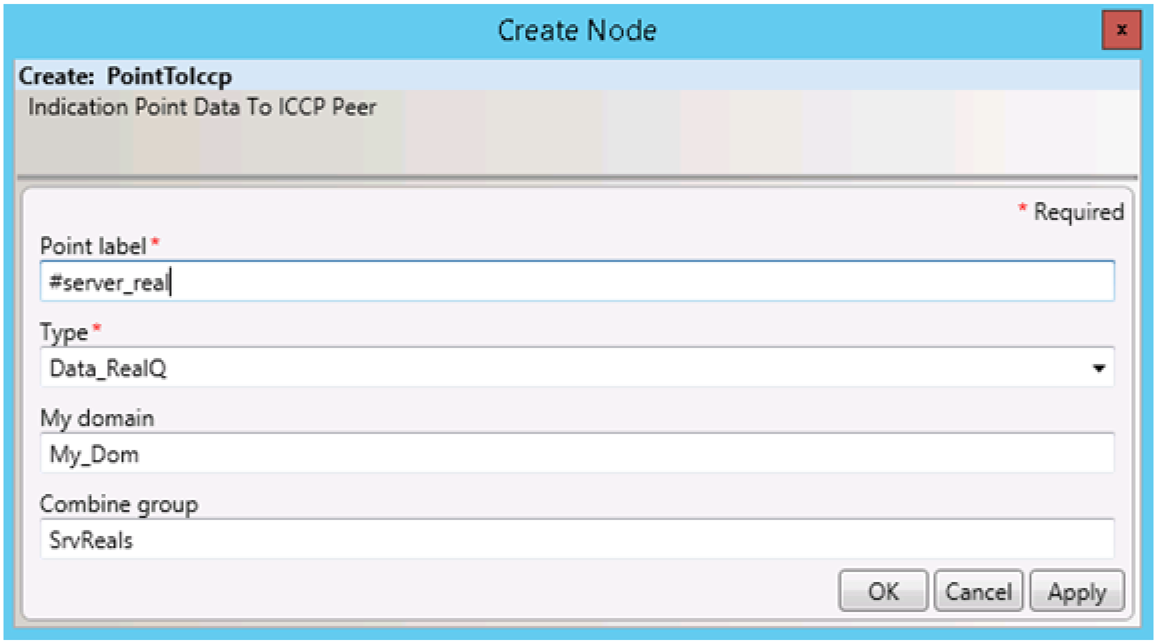
Create a Prototype Output Node within My_Vcc

This prototype output node defines a template for nodes that are able to pass data to the remote VCC His_Vcc.

1. Click on My_Vcc to select it.

2. Click on the Output tab at the bottom of the Templates panel.

3. Select the PointToIccp template. The following figure provides all of the information that you will need to fill in this PointToIccp form.



Create: PointToIccp

Indication Point Data To ICCP Peer

* Required

Point label *

#server_real

Type *

Data_RealQ

My domain

My_Dom

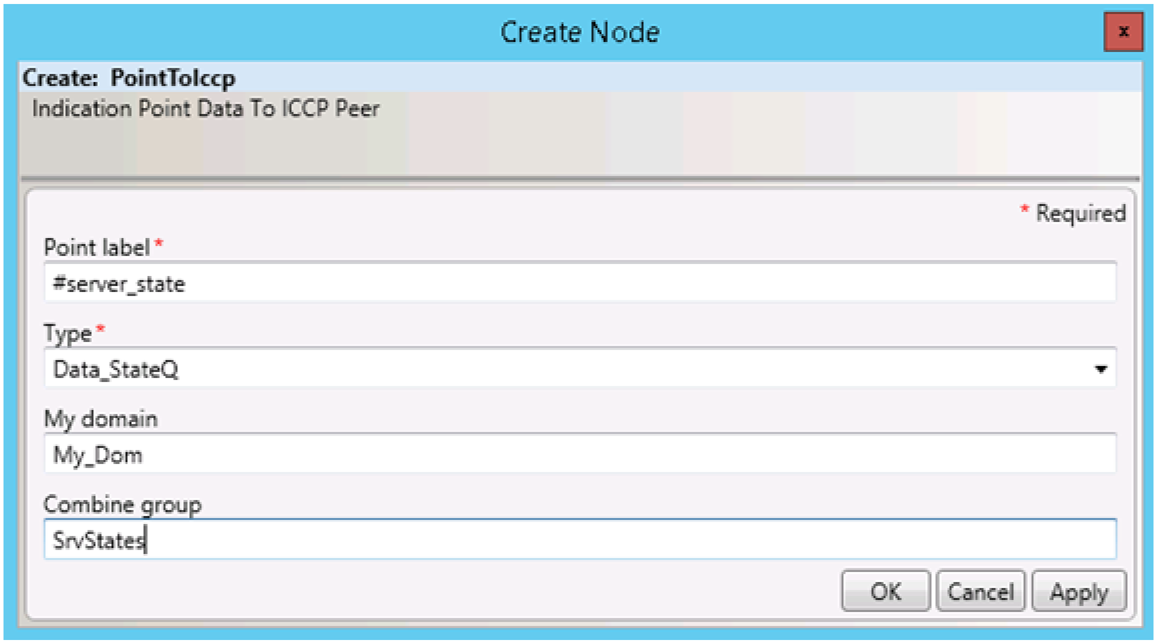
Combine group

SrvReals

4. Click **OK** to create this prototype node.

Create another Prototype Output Node within the My_Vcc Device

1. Create another prototype Output Node as you did for #server_real, except that this node is #server_state and has the type Data_StateQ. The following screenshot provides all the information that you need to fill in the form. Note that the data type is Data_StateQ.



Create: PointToIccp

Indication Point Data To ICCP Peer

* Required

Point label *

#server_state

Type *

Data_StateQ

My domain

My_Dom

Combine group

SrvStates

2. Click **OK** to create this prototype node.

When you have finished creating each of these nodes, click on the Node Table tab to see My_Vcc and all of its nodes. Figure 56 shows what they should look like.

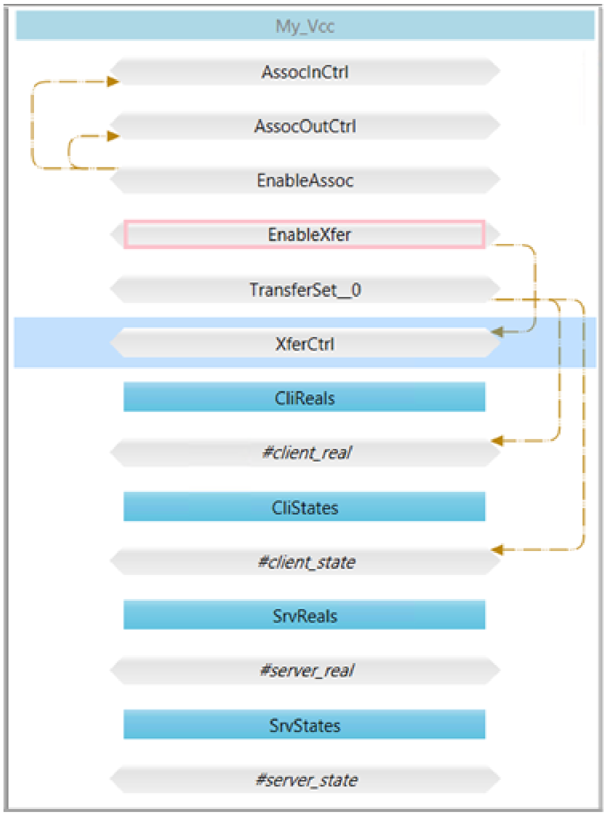


Figure 56: My_Vcc with Nodes

Create Nodes within the Processor Device

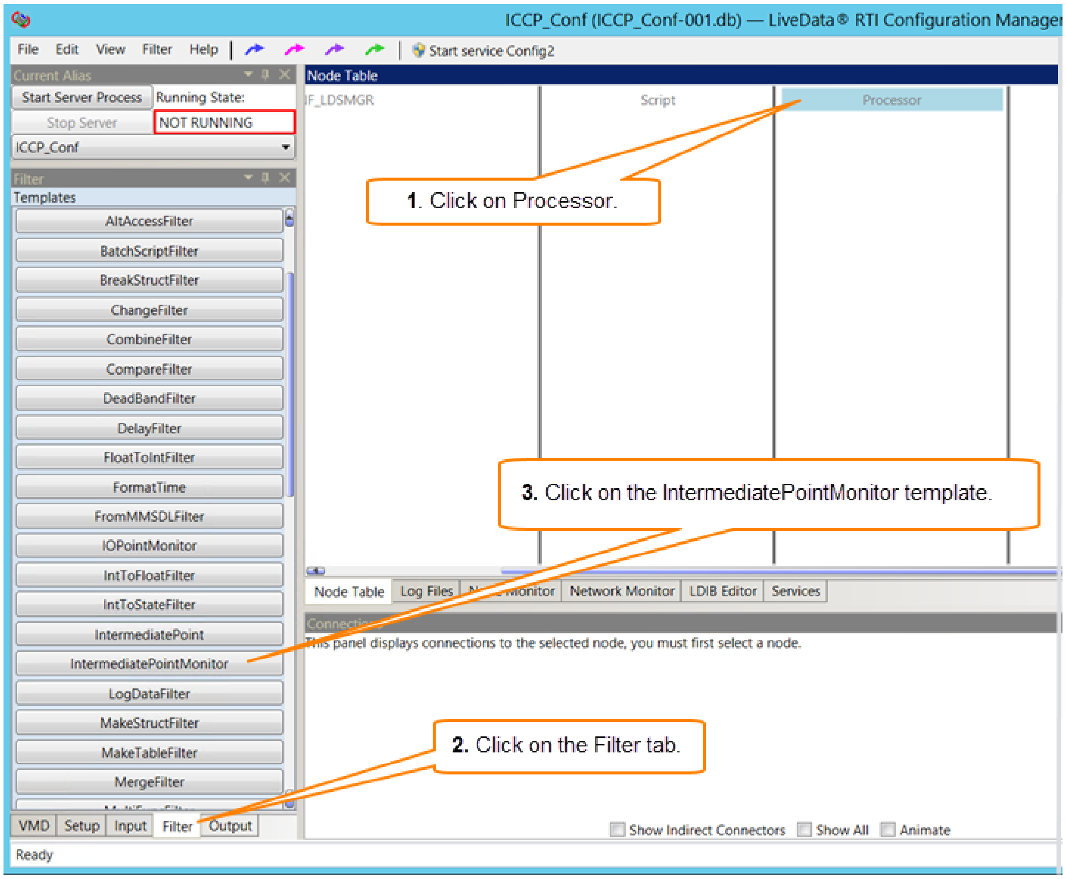
The Processor VMD in this sample configuration requires the creation of 7 IntermediatePointMonitor filter nodes. IntermediatePointMonitor nodes are storage locations that do not change the data. You can use these nodes to view the data that is passed to them from other nodes in the My_Vcc VMD and from nodes in the Script VMD.

The Processor nodes and their properties are shown in the following table. Figure 57 shows how to create an IntermediatePointMonitor node in the Processor VMD.

Point label/Node name	Type	Light	Combine Groups
AssocInCtrl	<integer:32>	Leave blank	Leave blank
AssocOutCtrl	<integer:32>	Leave blank	Leave blank
XferCtrl	<boolean>(<pack>16)	Leave blank	Leave blank
#client_real	Data_RealQ	Leave blank	CliReals
#client_state	Data_StateQ	Leave blank	CliStates
#server_real	Data_RealQ	Leave blank	SrvReals
#server_state	Data_StateQ	Leave blank	SrvStates

Table 1: Processor Node Properties

Create AssocInCtrl



This panel displays connections to the selected node, you must first select a node.

1. Click on Processor.

2. Click on the Filter tab.

3. Click on the IntermediatePointMonitor template.

Figure 57: Create an IntermediatePointMonitor Node

1. Click on Processor in the Node Table. A transparent blue bar will be displayed over the VMD to indicate that this device has been selected.

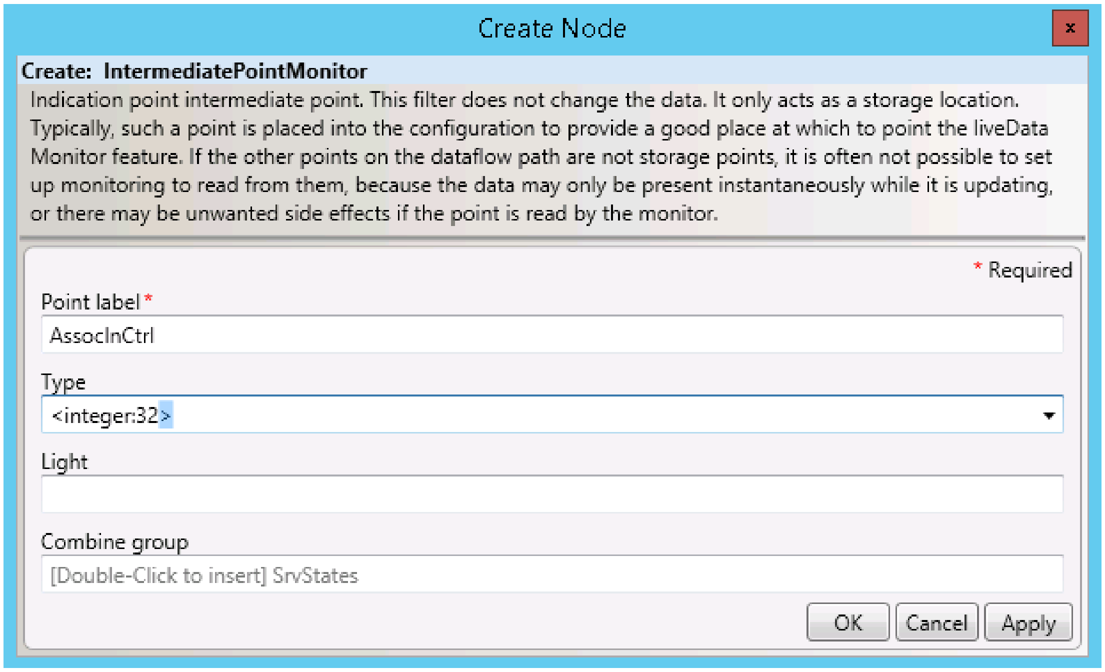
2. Click on the Filter tab at the bottom of the Templates panel.

3. Click on the IntermediatePointMonitor template.

4. Note that this type of filter does not change the data. It acts as a storage location. An IntermediatePointMonitor node serves as a place in memory where you can monitor the value of the data as it passes through the Processor.

If the other nodes in the dataflow are not storage points, it is often not possible to monitor them. Many nodes are not monitorable because the data is only present for a fleeting moment while the point is updating, or because there may be unwanted side effects if the point is read by a monitor.

The Create Node form is displayed in Figure 58.



Create Node

Create: IntermediatePointMonitor

Indication point intermediate point. This filter does not change the data. It only acts as a storage location. Typically, such a point is placed into the configuration to provide a good place at which to point the liveData Monitor feature. If the other points on the dataflow path are not storage points, it is often not possible to set up monitoring to read from them, because the data may only be present instantaneously while it is updating, or there may be unwanted side effects if the point is read by the monitor.

* Required

Point label *: AssocInCtrl

Type: <integer:32>

Light:

Combine group: [Double-Click to insert] SrvStates

Figure 58: IntermediatePointMonitor Node Form

1. Specify the Point label as AssocInCtrl for the purposes of this tutorial.

2. Under the Type label, specify <integer: 32> for the data type.

3. Leave out the value under Light.

4. Leave out the field beneath Combine group.

5. Note the description at the top of the form. It provides a definition of IntermediatePointMonitor nodes and why they are needed in configurations. You can find these descriptions at the top of each template form as you create the node and on the Properties panel on the right side of the page if you select the node in your network.

6. Click on the **OK** button on the bottom of the form.

Create AssocOutCtrl and XferCtrl

These IntermediatePointMonitor nodes are created using the properties displayed in Table 2.

Point label/Node name	Type	Light	Combine Groups
AssocInCtrl	<integer:32>	Leave blank	Leave blank
AssocOutCtrl	<integer:32>	Leave blank	Leave blank
XferCtrl	<boolean>(<pack>16)	Leave blank	Leave blank

Table 2: AssocInCtrl, AssocOutCtrl, and XferCtrl Node Properties

1. As you can see, AssocOutCtrl has the same type as AssocInCtrl. Create this node as you created AssocInCtrl.

2. XferCtrl is similar to the other Processor nodes except that it has a different data type. Create this IntermediatePointMonitor node, specifying the type as <boolean>(<pack>16).

Create Processor Prototype Nodes

In this section, you will create four prototype nodes that are based on the IntermediatePointMonitor template. Each prototype node begins with a pound sign (#). Unlike the other nodes, each of these nodes has a specified combine group as shown in the following table.

Point label/Node name	Type	Light	Combine Groups
#client_real	Data_RealQ	Leave blank	CliReals
#client_state	Data_StateQ	Leave blank	CliStates
#server_real	Data_RealQ	Leave blank	SrvReals
#server_state	Data_StateQ	Leave blank	SrvStates

Table 3: Processor Prototype Node Properties

Create these nodes, assigning the properties shown in Table 3.

Remember to use the IntermediatePointMonitor template that is found in the Filter tab of the Templates panel.

Create Nodes within the Script VMD

Create Setup Nodes

In order for the script to work, you need to create four Setup nodes.

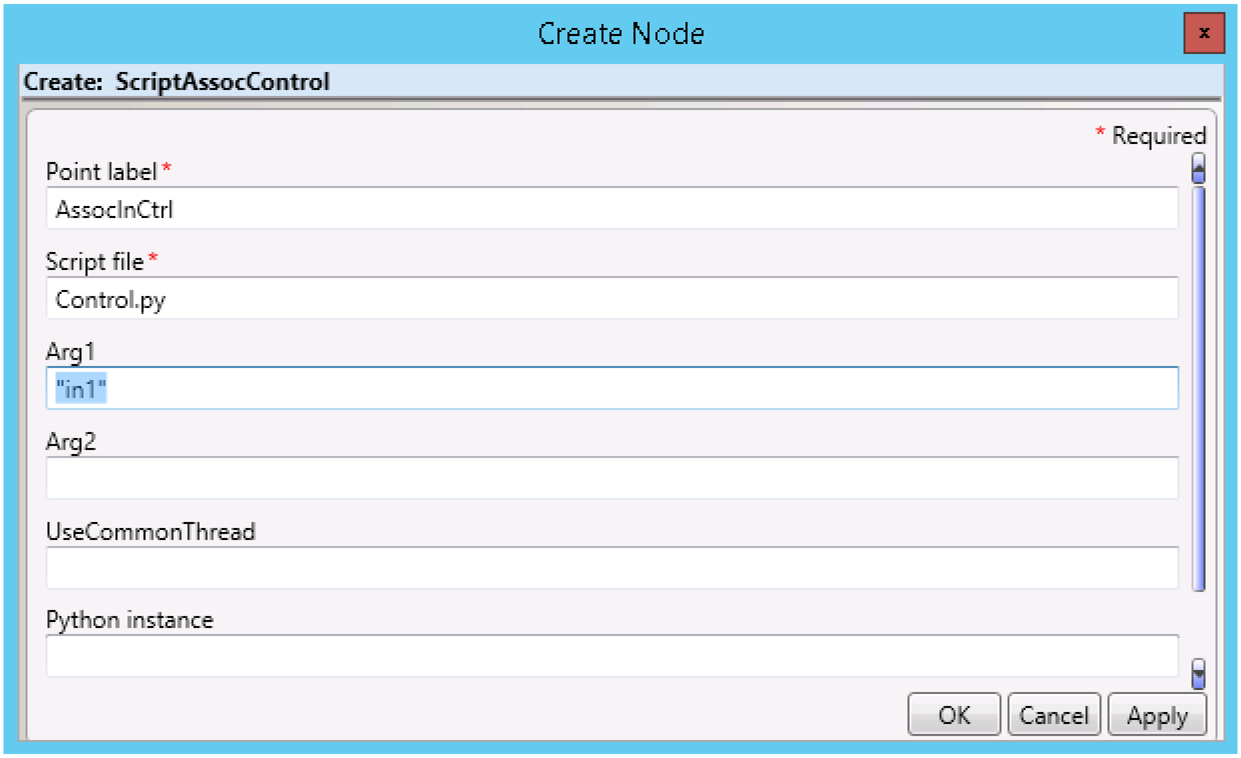
AssocInCtrl and AssocOutCtrl are ScriptAssocControl nodes that monitor ICCP inbound and outbound associations, respectively.

Volts is a SetupScript node that sets up the Script Agent, which feeds data into the Processor nodes; the Processor nodes then feed data into My_Vcc nodes.

ScriptTransferControl controls and monitors the ICCP Transfer Set identified by the name TransferSet_0.

To create AssocInCtrl:

1. Select the Script VMD.
2. Select the Setup tab beneath the Templates panel.
3. Select the ScriptAssocControl template to display the Create Node form.
4. Specify AssocInCtrl for the Point label.
5. Specify Control.py for the Script file.
6. Specify "in1" in the Arg1 field.
7. Click **OK** to create the node.



Point label * AssocInCtrl

Script file * Control.py

Arg1 "in1"

Arg2

UseCommonThread

Python instance

To create AssocOutCtrl:

1. Go through the same steps as you did to create AssocInCtrl, changing only the Point label and the value of Arg1.
2. Select the Setup tab beneath the Templates panel.
3. Select the ScriptAssocControl template to display the Create Node form.
4. Specify AssocOutCtrl for the Point label.
5. Specify Control.py for the Script file.
6. Specify "out1" in the Arg1 field.
7. Click **OK** to create the node.

To create Volts:

1. Select the Setup tab beneath the Templates panel.
2. Select the SetupScript template to display the Create Node form.
3. Specify Volts for the Script label.
4. Specify Volts.py for the Script file.
5. Specify the period as 10000.
6. Leave all of the other fields blank.
7. Click **OK** to create the node.

To create XferCtrl:

1. Select the Setup tab beneath the Templates panel.
2. Select the ScriptTransferControl.
3. Specify <boolean>(<pack>16) as the Type.
4. Specify Control.py as the script file.
5. Specify "ts1" as Arg1.
6. Leave all other fields blank.
7. Click **OK** to create this node.

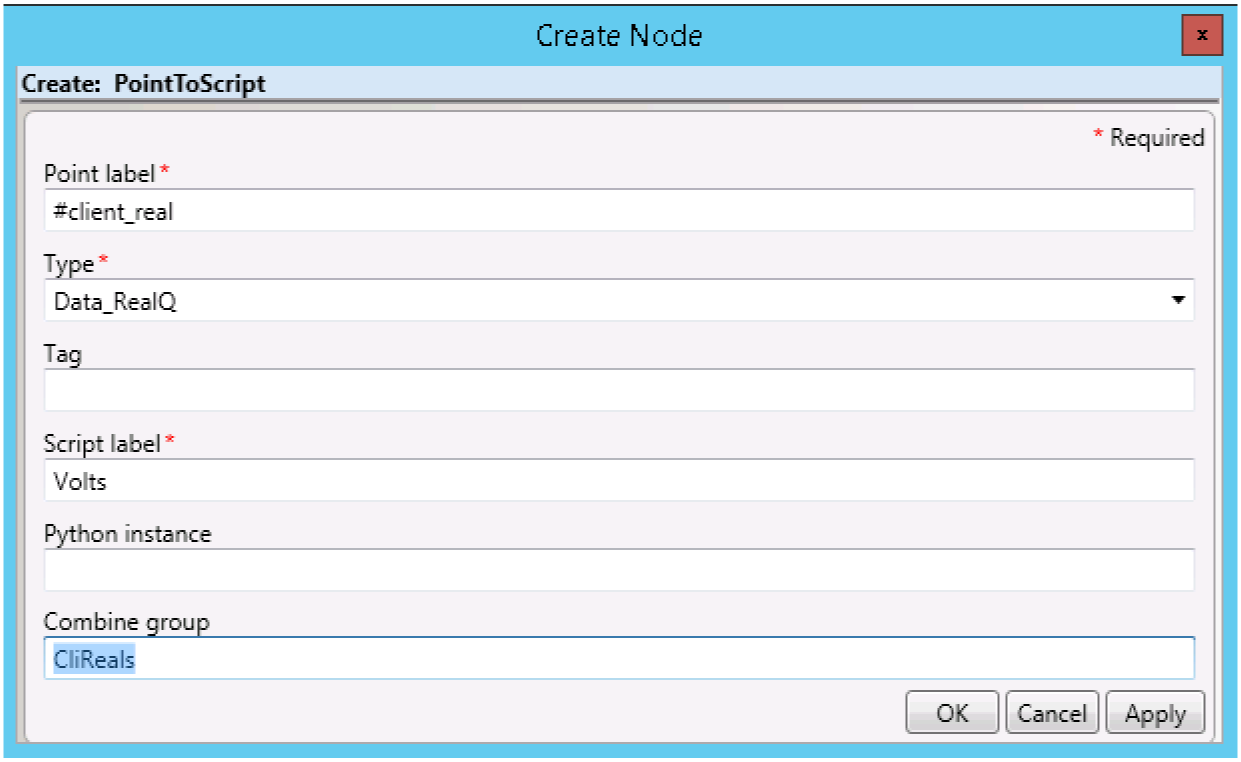
Create Prototype Nodes in the Script VMD

In this section, you will create four prototype nodes:

- Two PointToScript output nodes that write data from a remote VMD to the Volts script, and then stored in the IntermediatePointMonitor nodes of the Processor VMD.
- Two ScriptToPoint input nodes that write data from the Volts script to nodes in the Processor VMD.

Create PointToScript Prototype Nodes

1. Select the Script VMD.
2. Select the Output tab beneath the Templates panel.



Create: PointToScript

Point label *
#client_real

Type *
Data_RealQ

Tag

Script label *
Volts

Python instance

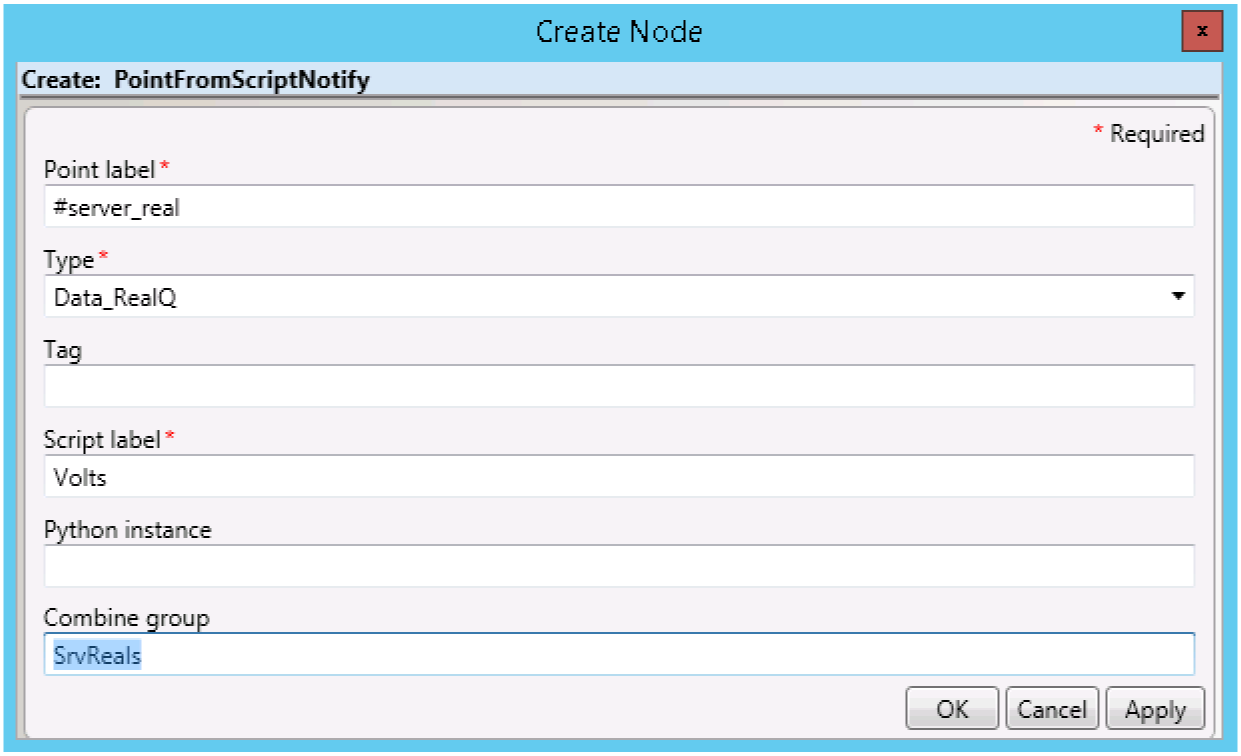
Combine group
CliReals

3. Select the PointToScript template.
4. Specify #client_real for the Point label.
5. Specify Data_RealQ for the Type.
6. Specify Volts for the Script label.
7. Specify CliReals for the Combine group.
8. Click **OK** to create the node.
9. Select PointToScript again to create another prototype node.
10. Specify #client_state for the Point label of the new node.
11. Specify Data_StateQ for the Type.
12. Specify Volts for the Script label.
13. Specify CliStates for the Combine group.
14. Click **OK** to create the node.

Create PointFromScriptNotify Nodes

PointFromScriptNotify nodes get data from the Volts.py script after receiving notification from the Volts node.

1. Select the Input tab beneath the Templates panel.
2. Select the PointFromScriptNotify template.



Point label * #server_real

Type * Data_RealQ

Tag

Script label * Volts

Python instance

Combine group SrvReals

3. Specify #server_real for the Point label.
4. Specify Data_RealQ for the Type.
5. Specify Volts for the Script label.
6. Specify SrvReals for the Combine group.
7. Click **OK** to create the node.
8. Select PointFromScriptNotify again to create another prototype node of this type.
9. Specify #server_state for the Point label.
10. Specify Data_StateQ for the Type.
11. Specify Volts for the Script label.
12. Specify SrvStates for the Combine group.
13. Click **OK** to create the node.

Create Connections between the Nodes

For the purposes of this tutorial, you will create two types of connectors between the nodes:

- Two-way connectors
- Update connectors

Figure 59 shows the Script VMD, the Processor VMD, and the My_Vcc VMD in addition to the nodes within them and the connections between these nodes.

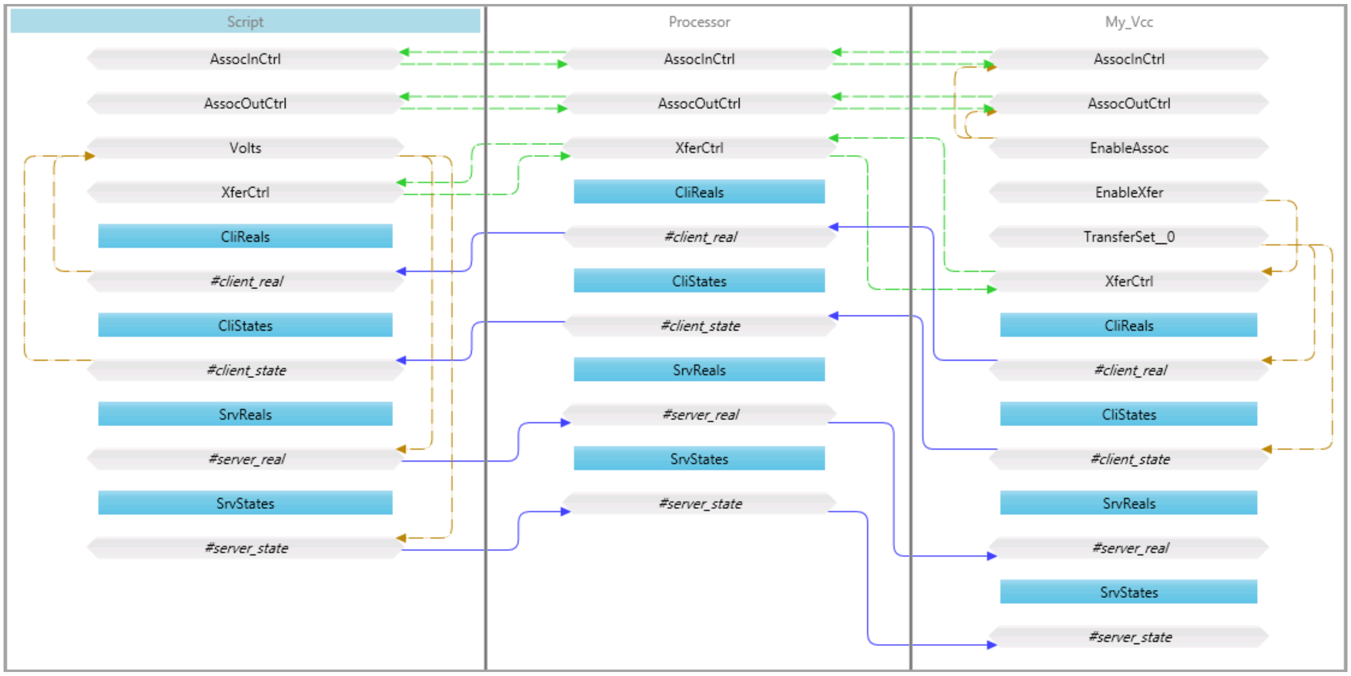


Figure 59: Node Table with Connectors

Before you begin, find the Connectors bar in the header of RTI Configuration Manager as shown below.

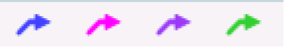


Figure 60: Connectors

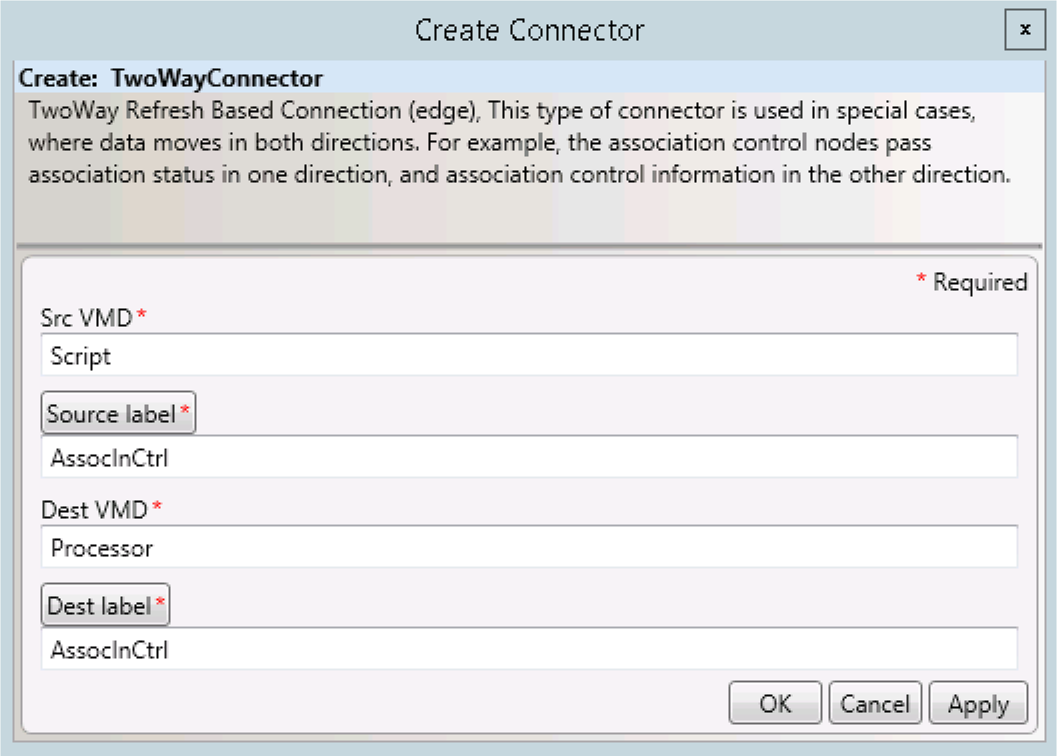
Two-Way Refresh Connections

The green arrow is a two-way refresh connector. This type of connector is used in special cases where data moves in both directions. There is a two-way refresh connection between AssocInCtrl in the Script VMD and AssocInCtrl of the Processor VMD; there is another two-way refresh connection between the AssocInCtrl node of the Processor VMD and the AssocInCtrl node of the My_Vcc VMD.

To create these connectors:

1. Click on the AssocInCtrl node of the Script VMD.
2. Select the green two-way connector in the header bar. Figure 61 shows the Create Connector form.

By selecting the node from which the connector originates, you save time because the Src VMD and Source label fields will be filled in for you.



Create Connector

Create: TwoWayConnector

TwoWay Refresh Based Connection (edge), This type of connector is used in special cases, where data moves in both directions. For example, the association control nodes pass association status in one direction, and association control information in the other direction.

* Required

Src VMD *: Script

Source label *: AssocInCtrl

Dest VMD *: Processor

Dest label *: AssocInCtrl

Figure 61: Two-Way Connector Form

3. RTI Configuration Manager will correctly fill in the Src VMD field with Script and the Source label field with AssocInCtrl.

4. Specify Processor in the Dest VMD field.

5. Specify AssocInCtrl in the Dest label field.

6. Click **OK** to create this connection.

7. Select the Processor VMD to start a new connection.

8. Select AssocInCtrl in the Processor VMD.

9. Select the green two-way connector in the header bar.

10. RTI Configuration Manager will correctly fill in the Src VMD field with Processor and the Source label field with AssocInCtrl.

11. Specify My_Vcc in the Dest VMD field and AssocInCtrl in the Dest label field.

12. Click **OK** to create this connection.

13. Repeat these steps to create a two-way connector between each of the following nodes:

- AssocOutCtrl in the Script VMD and AssocOutCtrl in the Processor VMD
- AssocOutCtrl in the Processor VMD and AssocOutCtrl in the My_Vcc VMD
- XferCtrl in the Script VMD and XferCtrl in the Processor VMD
- XferCtrl in the Processor VMD and XferCtrl in the My_Vcc VMD

Update-Based Connections

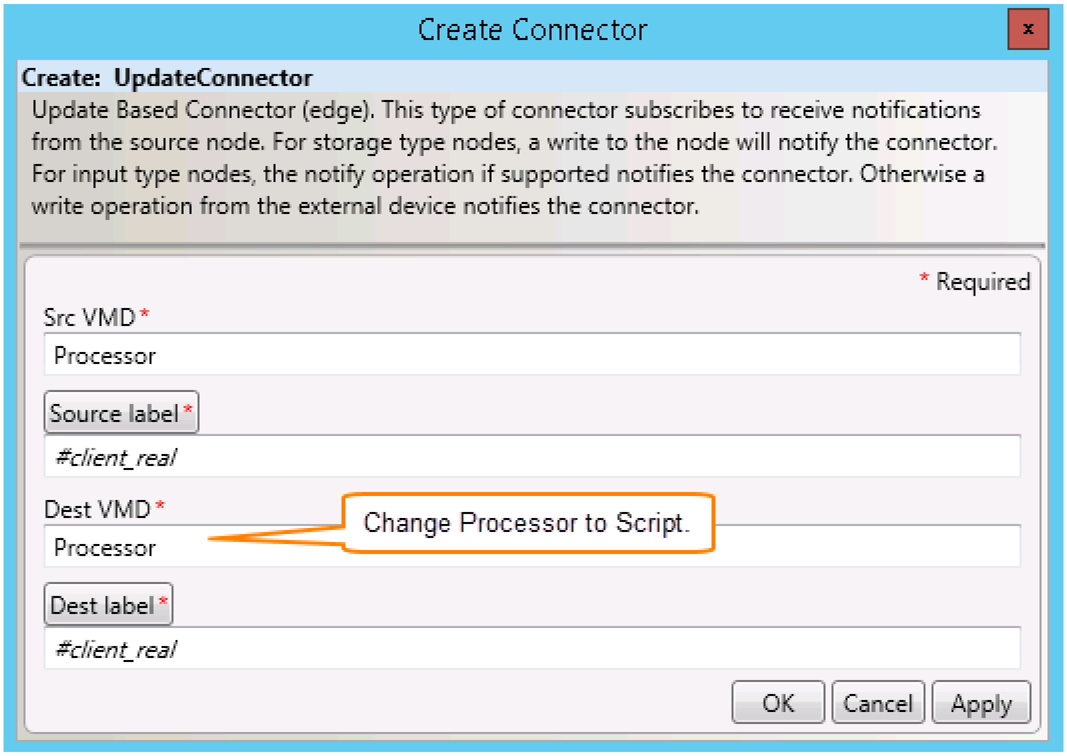
Update-based connectors (edge) subscribe to receive notifications from the source node. For storage type nodes, a write to the source node will notify the connector. For input type nodes, the notify operation, if supported, notifies the connector. Otherwise, a write operator from the other device notifies the connector.

Unlike Two-Way Connectors, you must select the node from which the connector originates first to show the direction of the Connector.

You need to create update-based connectors between the prototype nodes, starting with the connection between #client_real in the Processor VMD and #client_real in the Script VMD.

To create this connector:

1. Select the Processor VMD.
2. Select #client_real.
3. Select the blue arrow from the Connectors bar. Figure 62 shows the Create Connector form.



Create Connector

Create: UpdateConnector

Update Based Connector (edge). This type of connector subscribes to receive notifications from the source node. For storage type nodes, a write to the node will notify the connector. For input type nodes, the notify operation if supported notifies the connector. Otherwise a write operation from the external device notifies the connector.

* Required

Src VMD *
Processor

Source label *
#client_real

Dest VMD *
Processor

Change Processor to Script.

Dest label *
#client_real

Figure 62: Update Connector Form

4. The Src VMD field is correctly set to Processor.
5. The Source label is correctly set to #client_real.
6. Change the Dest VMD to Script.
7. The Dest label is correctly set to #client_real.
8. Click **OK** to create the connector.
9. Create the update-based connectors for the configuration as shown in the following table.

Source VMD	Source Label	Dest VMD	Dest Label
My_Vcc	#client_real	Processor	#client_real
My_Vcc	#client_state	Processor	#client_state
Processor	#client_state	Script	#client_state
Script	#server_real	Processor	#server_real
Processor	#server_real	My_Vcc	#server_real

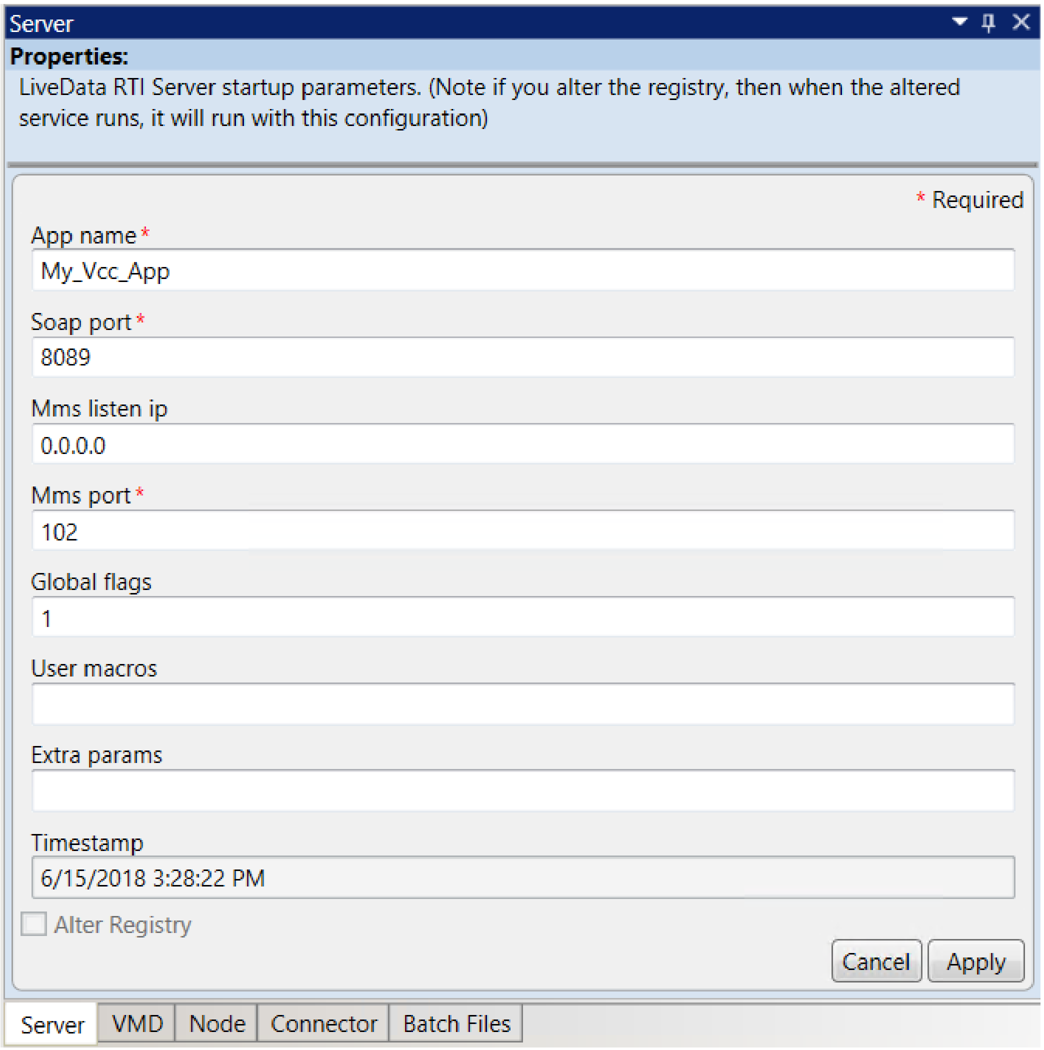
Script	#server_state	Processor	#server_state
Processor	#server_state	My_Vcc	#server_state

10. Compare your own Node Table with the configuration shown in "Node Table with Connectors" on page 92 to ensure that the two configurations match.

Specify the Properties of RTI Server

Before completing the configuration, you need to specify the RTI Server startup parameters.

1. Click on the Server tab at the bottom of the Properties panel on the right side of RTI Configuration Manager as shown in Figure 63.



Server

Properties:

LiveData RTI Server startup parameters. (Note if you alter the registry, then when the altered service runs, it will run with this configuration)

* Required

App name*: My_Vcc_App

Soap port*: 8089

Mms listen ip: 0.0.0.0

Mms port*: 102

Global flags: 1

User macros:

Extra params:

Timestamp: 6/15/2018 3:28:22 PM

□ Alter Registry

Figure 63: RTI Server Startup Parameters

2. Specify My_Vcc_App in the App name field.

Note: Some of the fields specified in Steps 3 through 6 might be filled in already. If the fields are set to the same values specified in Steps 3 through 5, you will not need to change them.

3. Specify 8089 in the Soap port field. The Soap port is the TCP port that RTI Server uses for SOAP communications.

4. Specify 0.0.0.0 in the Mms listen ip field. The MMS Listen IP is the TCP IP address RTI Server uses for MMS communications.

5. Specify 102 in the Mms port field. The MMS port is the TCP port that RTI Server uses for MMS communications.

6. Specify 1 in the Global flags field. 1 indicates that RTI Server is not configured to run Secure ICCP. In the future, if you would like to configure RTI Server to run Secure ICCP, you would specify 3. However, there are additional installation and configuration issues that you need to address prior to using Secure ICCP.

7. If you had to change the SOAP port, MMS port, MMS listen IP, or Global flags, then you need to check **Altar Registry** at the bottom of the form so that your changes are saved to the service associated with your alias.

8. Click on the **Apply** button.

Load a Batch File

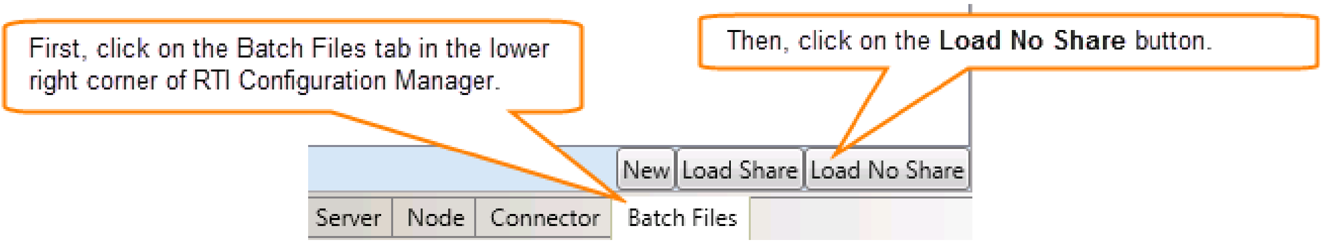
Once you have created a configuration, you can load batch files that create the hundreds or thousands of actual points in a network, allowing you to trace and monitor the dataflow from any point. A batch file is a .csv (comma-separated variable) file that provides values for each of the properties specified in the prototype nodes. There is a mapping between the prototype nodes that you defined and the batch file nodes that you will load.

RTI Configuration Manager provides a batch file for this ICCP configuration. The batch file fills in the details that distinguish one node from another. In this simple example, the batch file provides only the point label. However, in production a batch file would fill in values for many more fields. Figure 64 shows you the contents of this batch file, named IccpScriptTemplate.

```
! Protocol Server / RTI Example Batch File
! for use with the configuration in ScriptTemplate.db
!
! Copyright (c) 2007 LiveData, Inc.  All rights reserved.
#client_real
ClientReal_0
ClientReal_1

#client_state
ClientState_0
ClientState_1

#server_real
ServerReal_0
ServerReal_1

#server_state
ServerState_0
ServerState_1
```

Figure 64: Contents of the IccpScriptTemplate Batch File

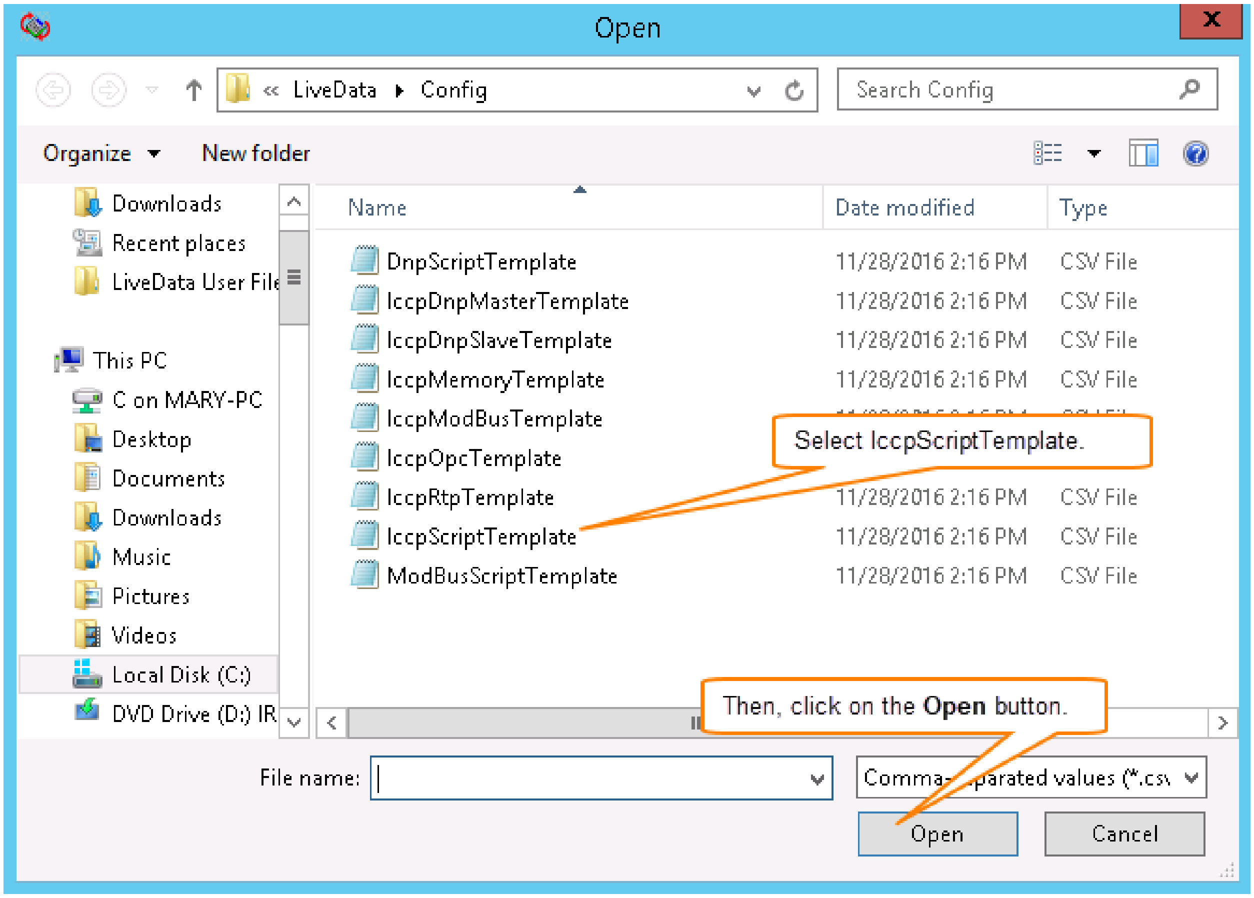
To load IccpScriptTemplate:



First, click on the Batch Files tab in the lower right corner of RTI Configuration Manager.

Then, click on the **Load No Share** button.

1. Click on the Batch Files tab, which is beneath the Properties panel on the right side of RTI Configuration Manager.
2. Click on the **Load No Share** button. Clicking on **Load No Share** displays a directory of batch files in Windows Explorer.



Name	Date modified	Type
DnpScriptTemplate	11/28/2016 2:16 PM	CSV File
IccpDnpMasterTemplate	11/28/2016 2:16 PM	CSV File
IccpDnpSlaveTemplate	11/28/2016 2:16 PM	CSV File
IccpMemoryTemplate	11/28/2016 2:16 PM	CSV File
IccpModBusTemplate	[illegible]	[illegible]
IccpOpcTemplate	[illegible]	[illegible]
IccpRtpTemplate	11/28/2016 2:16 PM	CSV File
IccpScriptTemplate	11/28/2016 2:16 PM	CSV File
ModBusScriptTemplate	11/28/2016 2:16 PM	CSV File

File name:

Comma-separated values (*.csv

Select IccpScriptTemplate.

Then, click on the **Open** button.

Figure 65: Load IccpScriptTemplate.csv into the ICCP Configuration

3. Select IccpScriptTemplate as shown in Figure 65.

4. Click on the **Open** button, which will generate nodes for each of the prototype nodes that you created. Figure 66 shows how the configuration looks with the generated batch nodes.

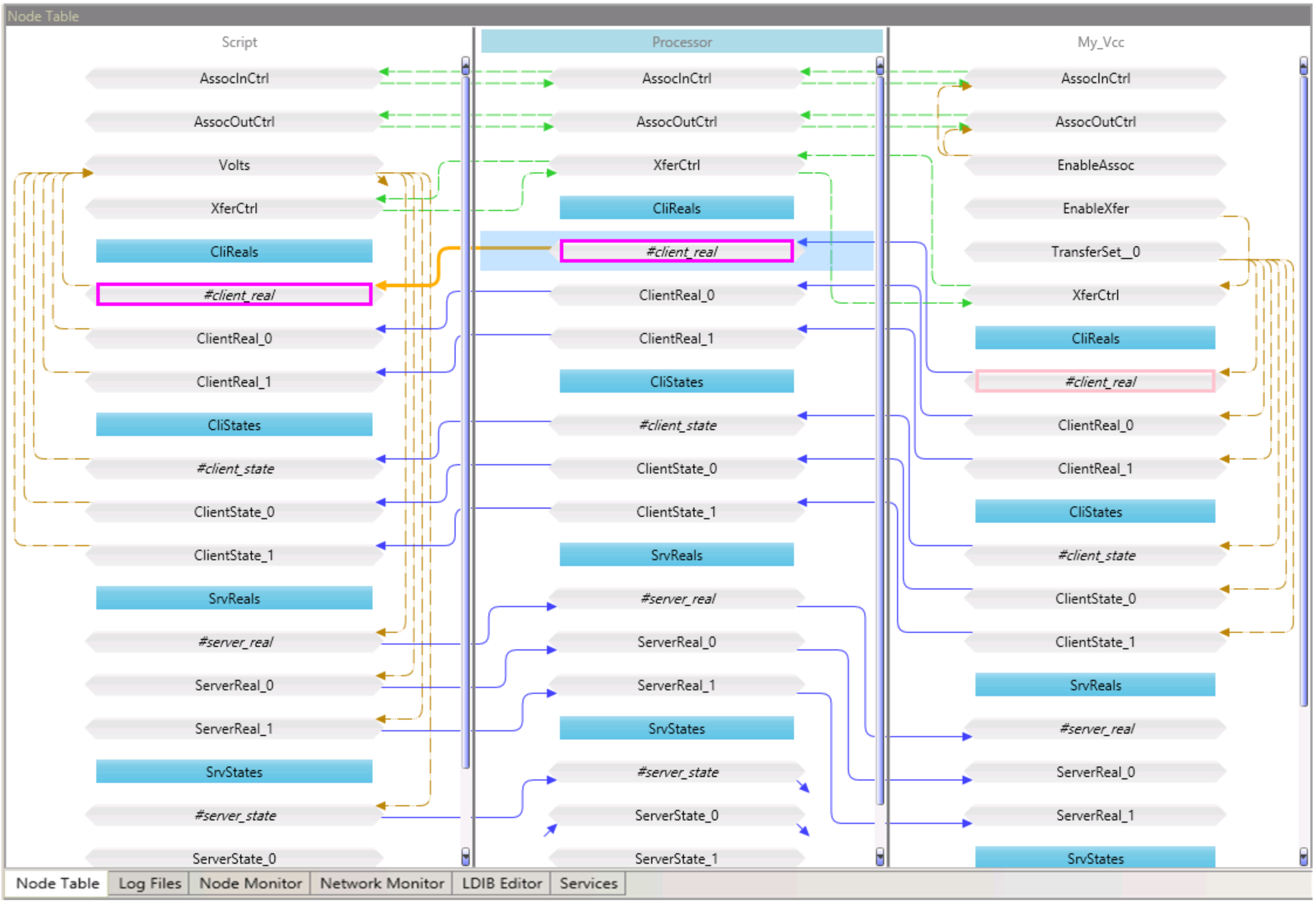


Figure 66: Node Table of Complete Configuration with the Batch File Loaded

Whenever you or a LiveData Utilities Professional Services engineer makes changes to a batch file to add or delete a node or nodes, you will need to reload the batch file. Learning how to edit batch files is extremely useful. For more information on batch files, read *Batch Tutorial*.

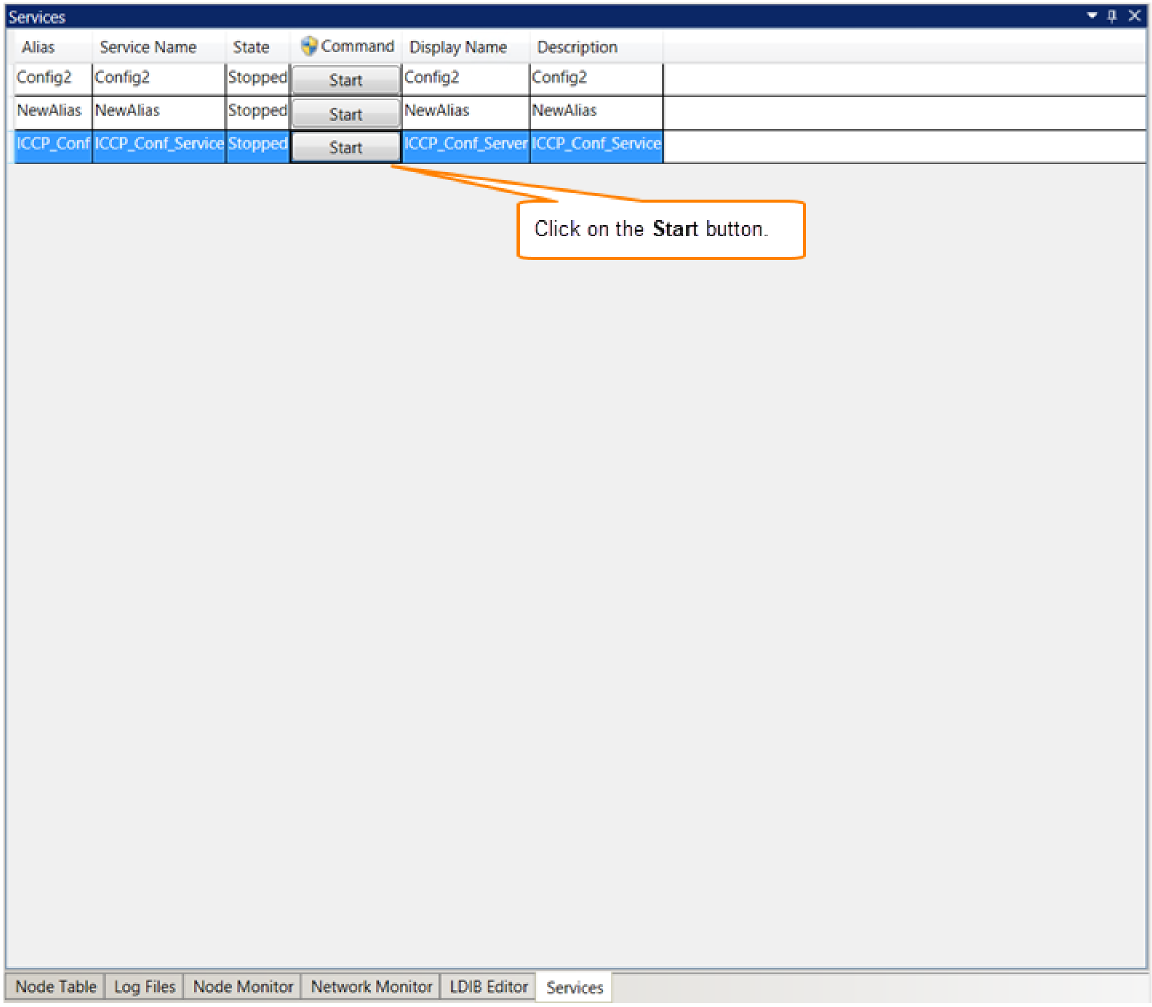
Start and Monitor the ICCP Configuration

Now that you have created one side of the mirrored ICCP configurations, you can start the configuration and examine the contents of each node in the Node Monitor.

Start RTI Server as a Service

To start RTI Server:

1. Click on the **Services** tab below the Node Table panel.
2. Click on the **Start** button in the ICCP_Conf row as shown below.

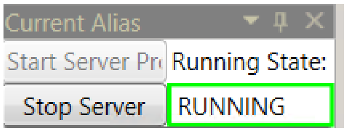


Services

Alias	Service Name	State	Command	Display Name	Description
Config2	Config2	Stopped	Start	Config2	Config2
NewAlias	NewAlias	Stopped	Start	NewAlias	NewAlias
ICCP_Conf	ICCP_Conf_Service	Stopped	Start	ICCP_Conf_Server	ICCP_Conf_Service

Click on the **Start** button.

After starting RTI Server, you will see that the Running State section (to the left of the Services panel) indicates RTI Server is running.

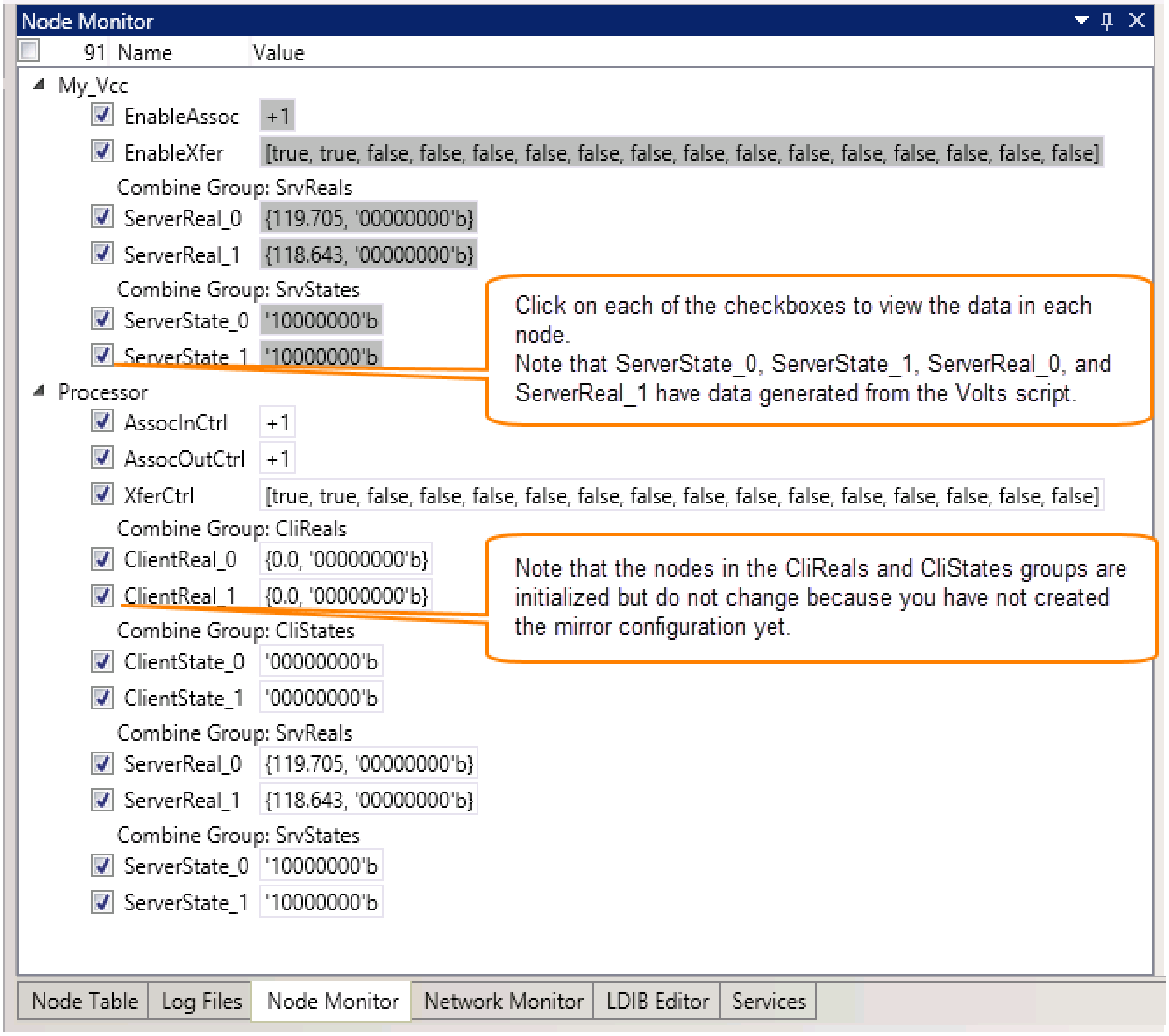


Current Alias

Start Server Pr	Running State:
Stop Server	RUNNING

Monitor the Dataflow

To monitor the data that flows through your configuration, click on the Node Monitor tab below the Services panel.



Node Monitor

□ 91 Name	Value
My_Vcc	
☑ EnableAssoc	+1
☑ EnableXfer	[true, true, false, false, false, false, false, false, false, false, false, false, false, false, false, false]
Combine Group: SrvReals	
☑ ServerReal_0	{119.705, '00000000'b}
☑ ServerReal_1	{118.643, '00000000'b}
Combine Group: SrvStates	
☑ ServerState_0	'10000000'b
☑ ServerState_1	'10000000'b
Processor	
☑ AssocInCtrl	+1
☑ AssocOutCtrl	+1
☑ XferCtrl	[true, true, false, false, false, false, false, false, false, false, false, false, false, false, false, false]
Combine Group: CliReals	
☑ ClientReal_0	{0.0, '00000000'b}
☑ ClientReal_1	{0.0, '00000000'b}
Combine Group: CliStates	
☑ ClientState_0	'00000000'b
☑ ClientState_1	'00000000'b
Combine Group: SrvReals	
☑ ServerReal_0	{119.705, '00000000'b}
☑ ServerReal_1	{118.643, '00000000'b}
Combine Group: SrvStates	
☑ ServerState_0	'10000000'b
☑ ServerState_1	'10000000'b

Click on each of the checkboxes to view the data in each node.
Note that ServerState_0, ServerState_1, ServerReal_0, and ServerReal_1 have data generated from the Volts script.

Note that the nodes in the CliReals and CliStates groups are initialized but do not change because you have not created the mirror configuration yet.

Figure 67: Start RTI Server and View the Data in Each Node

To monitor the configuration:

1. Click on each of the checkboxes next to the monitorable nodes in the My_Vcc and Processor VMDs. The nodes in the Script VMD are not displayed because they are not monitorable.

2. Notice the values changing in Server_Real_0, Server_Real_1, Server_State_0, and Server_State_1. The values in these nodes are generated by the Volts.py script. Whereas the values in Client_Real_0, Client_Real_1, Client_State_0, and Client_State_1 are initialized, yet never change because the remote configuration containing the VMD that will communicate with them is not created yet.

Note that you can change the values of the Processor nodes from the Node Monitor. For example, you can change AssocOutCtrl to 0. This does not have much impact now, but it will once an association has been made. A textbox around a value indicates that you can change the value. In Figure 66, each of the Processor nodes has a textbox around the value.

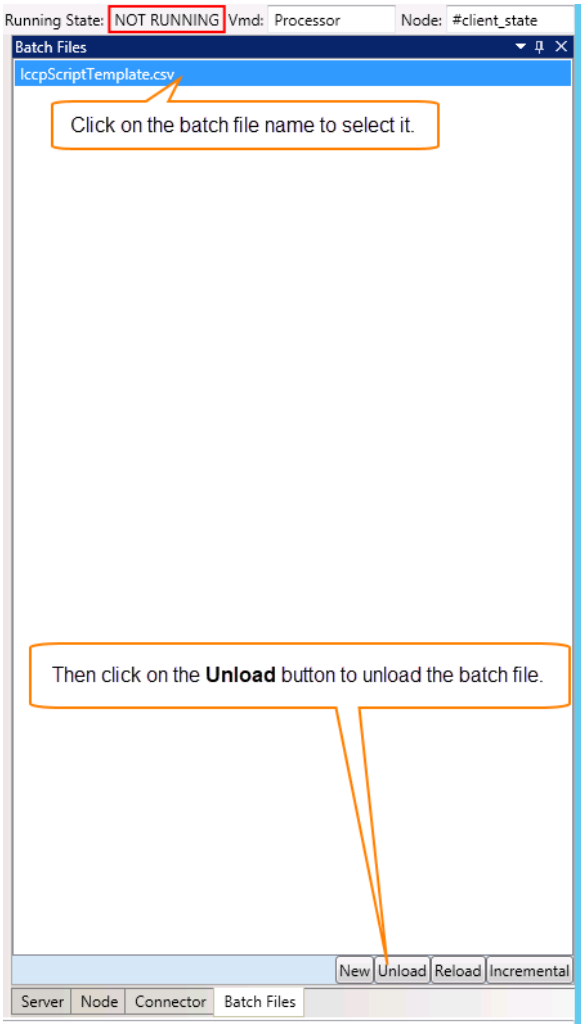
3. Stop RTI Server by selecting the **Services** tab beneath the Node Monitor panel, then click on the **Stop** button in the ICCP_Conf row. You will see the Running State change to Not Running.

Save and Import Your New Configuration

In this section, you will unload the batch file, save your new configuration, and import it into a different alias.

To unload the batch file

1. Click on the Batch Files tab beneath the Properties panel to display the Batch Files panel.



Running State: NOT RUNNING Vmd: Processor Node: #client_state

IccpScriptTemplate.csv

Click on the batch file name to select it.

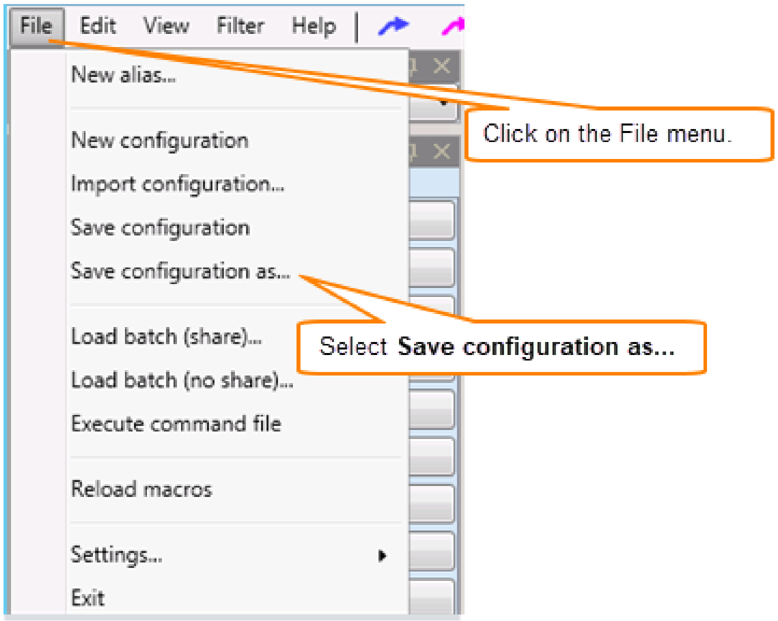
Then click on the **Unload** button to unload the batch file.

Figure 68: Batch Files Panel

2. Click on the IccpScriptTemplate.csv batch file to select it.
3. Click on the **Unload** button to unload it.

To Save the Configuration Database File

1. Click on the File menu.



Click on the File menu.

Select **Save configuration as...**

Figure 69: Save the Configuration as My_Vcc

2. Select **Save Configuration as...**

3. Save it as My_Vcc.db.

To Create a New Workspace

1. To orient yourself and organize your work, click on the File menu and select New Alias to create a new alias. An alias names a workspace for a new configuration on the same instance of RTI Server. Usually. each configuration is run on a separate instance of RTI Server that is installed on a different computer or VM. Your first configuration was created in the default alias named cfg.

2.

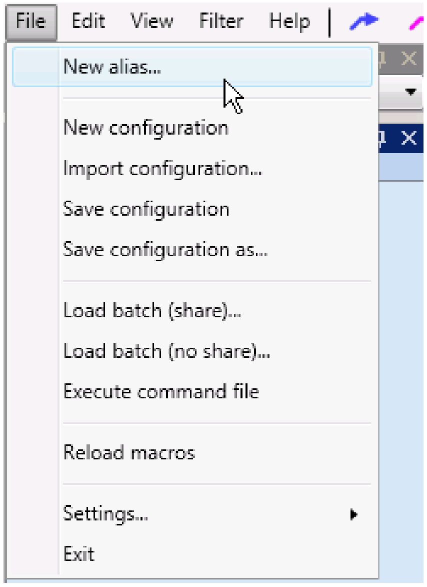
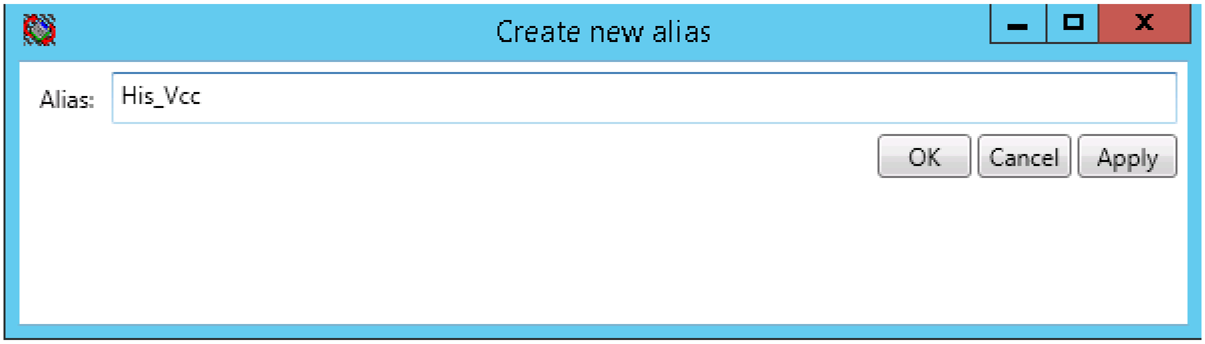


Figure 70: Create a New Alias

3. Select **New alias...** from within the File menu. Figure 16 displays the Create a new alias form.



Alias: His_Vcc

Figure 71: Name the New Alias His_Vcc

4. Specify His_Vcc in the Alias field.

5. Click on the **OK** button.

If the version of the loaded dataflow macros differs from the current dataflow macros, RTI Configuration Manager will issue the following message:

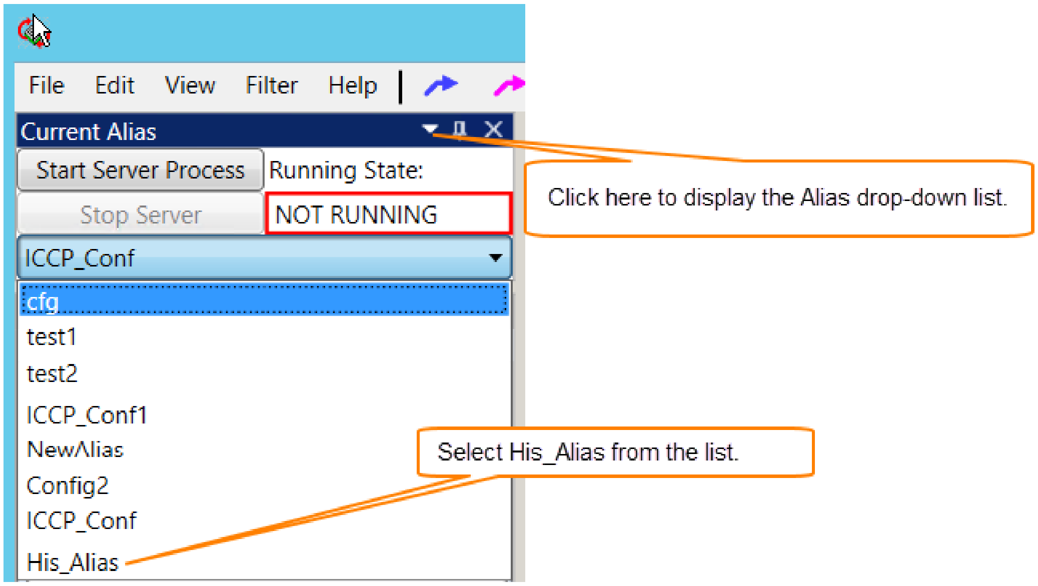
"The loaded DataflowMacros version does not match the DataflowMacros file version."

Whenever you see this message, click on **Reload macros** from the File menu.

To Import a Configuration

Now you will import the My_Vcc configuration into the His_Vcc configuration.

1. Under Current Alias in the top left panel, ensure that you have selected His_Vcc.



Running State: NOT RUNNING

Click here to display the Alias drop-down list.

Select His_Alias from the list.

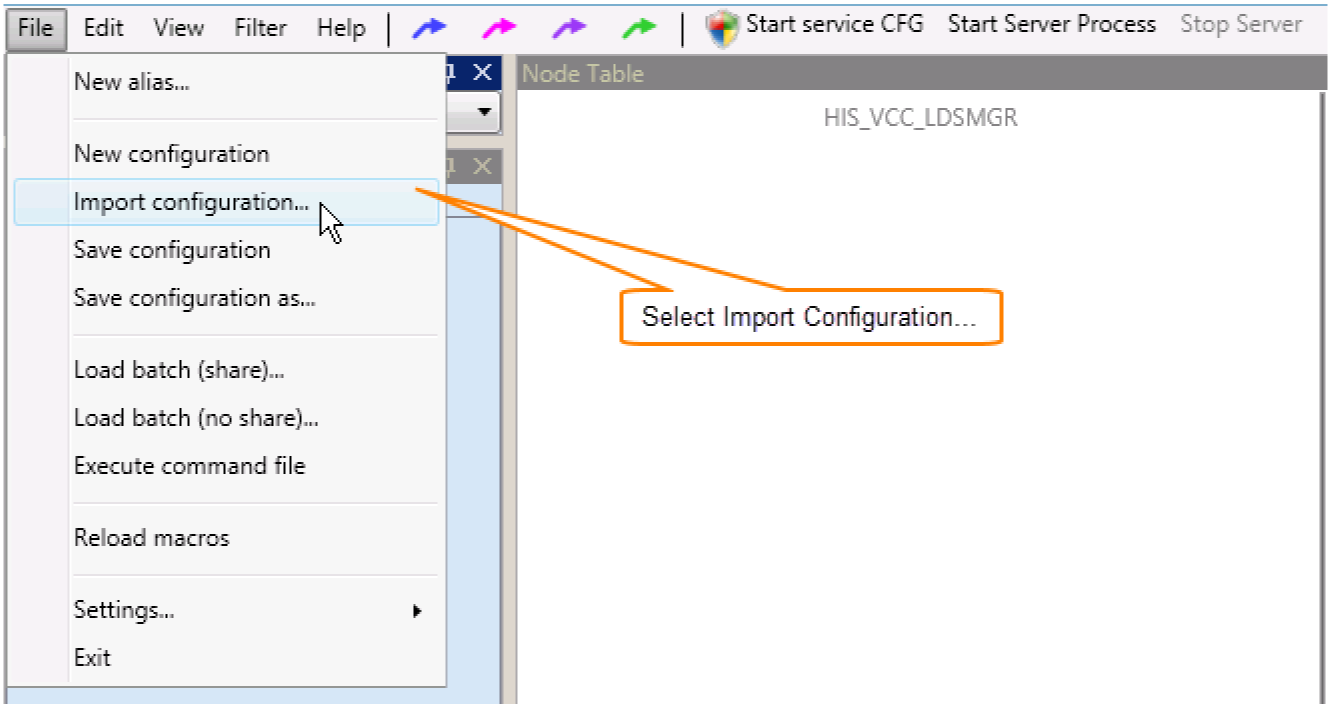
Figure 72: Aliases in Configurations List

2. If His_Vcc is not selected under Current Alias, then click on the arrow to see the drop-down list of aliases.

3. Click on His_Vcc to select it.

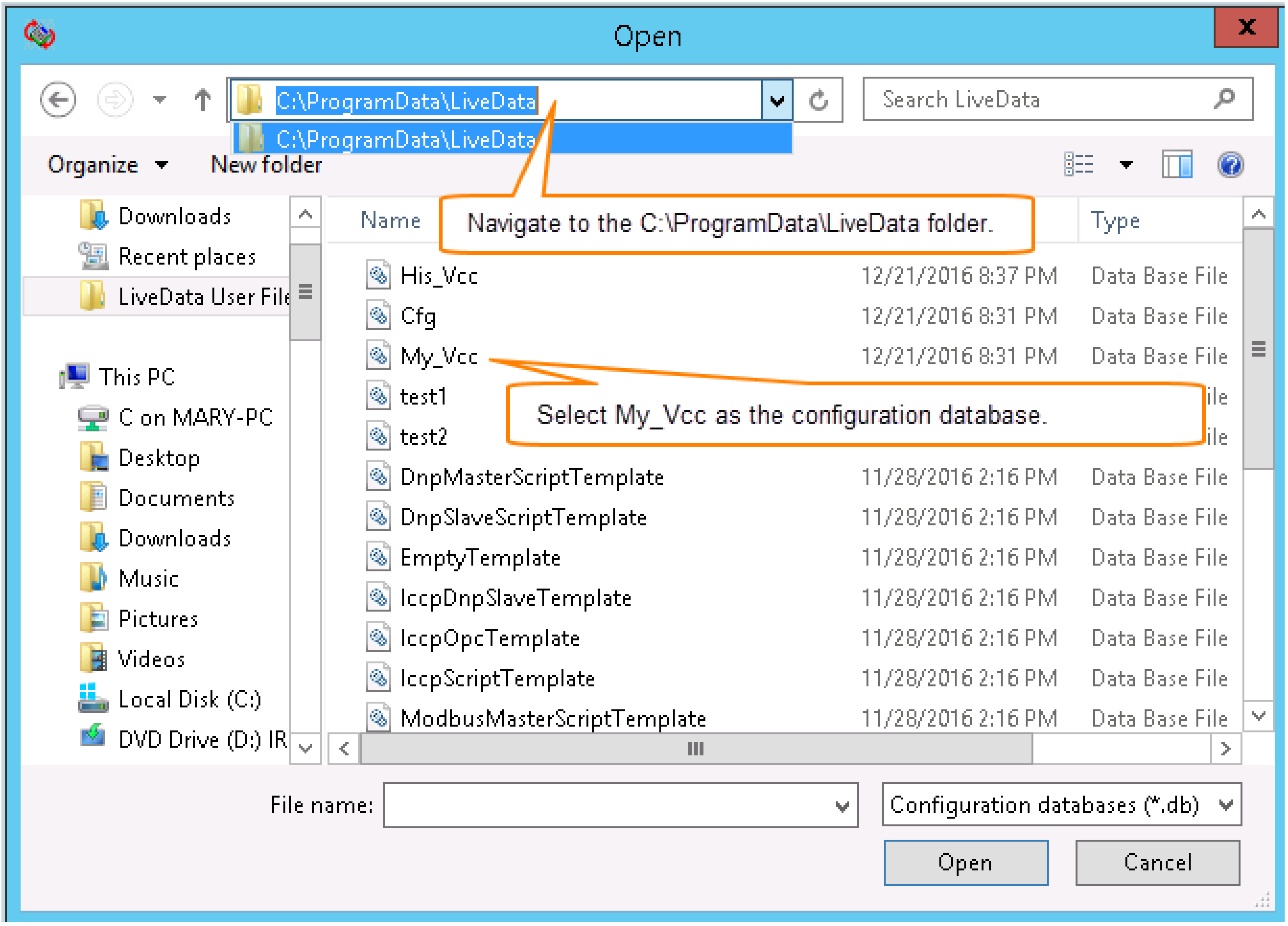
To Import a Configuration DB

Now you will import the configuration that you created in the previous sections into the His_Vcc alias.



Select Import Configuration...

1. Select **Import configuration...** from the File menu. Windows Explorer will be displayed.


Open
C:\ProgramData\LiveData
C:\ProgramData\LiveData
Search LiveData
Organize
New folder
Downloads
Recent places
LiveData User File
This PC
C on MARY-PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
DVD Drive (D:) IR
Name
Type
His_Vcc 12/21/2016 8:37 PM Data Base File
Cfg 12/21/2016 8:31 PM Data Base File
My_Vcc 12/21/2016 8:31 PM Data Base File
test1
test2
DnpMasterScriptTemplate 11/28/2016 2:16 PM Data Base File
DnpSlaveScriptTemplate 11/28/2016 2:16 PM Data Base File
EmptyTemplate 11/28/2016 2:16 PM Data Base File
IccpDnpSlaveTemplate 11/28/2016 2:16 PM Data Base File
IccpOpcTemplate 11/28/2016 2:16 PM Data Base File
IccpScriptTemplate 11/28/2016 2:16 PM Data Base File
ModbusMasterScriptTemplate 11/28/2016 2:16 PM Data Base File
File name:
Configuration databases (*.db)
Open
Cancel

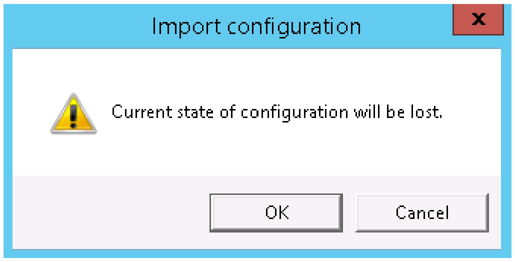
Navigate to the C:\ProgramData\LiveData folder.

Select My_Vcc as the configuration database.

2. Navigate to C:\ProgramData\LiveData.

3. Select My_Vcc.db to be imported.

4. Click on the **Open** button at the bottom of the form to start importing this configuration database. You will receive a message indicating that the current state of your configuration will be lost. Click on the **OK** button to dismiss it.



Import configuration

Current state of configuration will be lost.

Do not let this message worry you because there is no information in His_Vcc as of yet, since it is a new alias.

This import will bring in your My_Vcc configuration without your batch nodes.

Create a Mirror Configuration of My_Vcc

In this section, you will create a mirror configuration of My_Vcc. For the purposes of this tutorial, the mirror image will be called His_Vcc.

To Create a Mirror Image of the My_Vcc VMD

You need to change the Servers, VMD, and node properties.

This section is divided into the following topics:

- Change RTI Server Properties
- Change the My_Vcc VMD into His_Vcc
- Change #client_real to #server_real and #client_state to #server_state in His_Vcc
- Change #client_real to #server_real and #client_state to #server_state in the Script VMD
- Change #client_real to #server_real and #client_state to #server_state in the Processor VMD
- Check Network Addresses in the LDIB Editor
- Load a Batch File
- Start the RTI Server Process
- Start the First Configuration that has My_Vcc

Change RTI Server Properties

1. Click on the Server tab beneath the Properties panel.

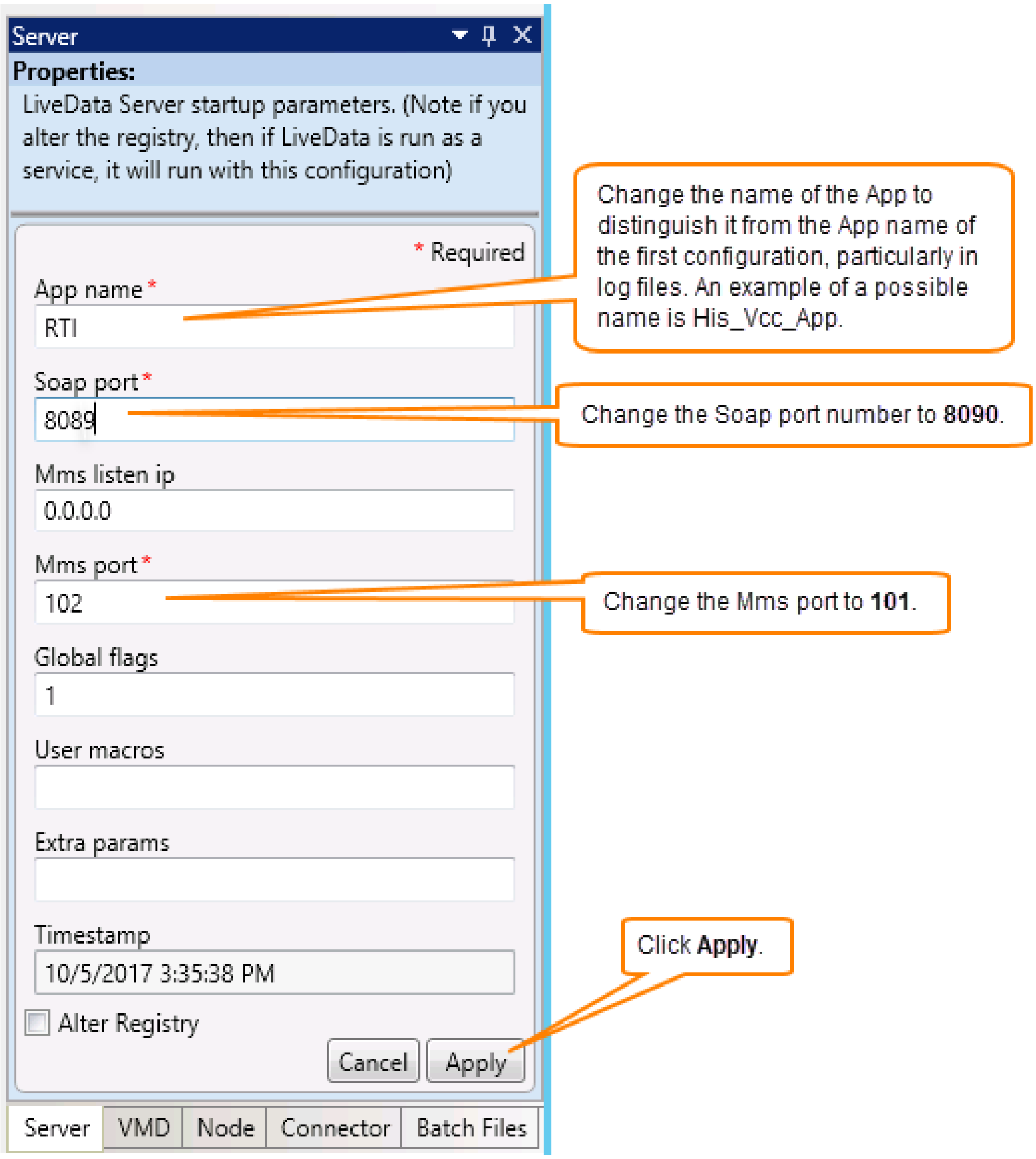
2. Change the App Name to a name that associates it with the mirror configuration, for example, His_Vcc_App. If you need to debug this configuration by looking at log files, any logs originating from this configuration will begin with the name you have given here.

3. Change the SOAP port number to 8090.

4. Change the MMS port number to 101.

5. Do not click on the **Alter Registry** checkbox because you will be starting this configuration as a process. Although you would never start a configuration that is in a production environment as a process, you might start a second configuration as a process if you are using the second configuration solely to test your main configuration.

6. Click on the **Apply** button.



Server

Properties:

LiveData Server startup parameters. (Note if you alter the registry, then if LiveData is run as a service, it will run with this configuration)

* Required

App name*: RTI

Soap port*: 8089

Mms listen ip: 0.0.0.0

Mms port*: 102

Global flags: 1

User macros:

Extra params:

Timestamp: 10/5/2017 3:35:38 PM

□ Alter Registry

Change the name of the App to distinguish it from the App name of the first configuration, particularly in log files. An example of a possible name is His_Vcc_App.

Change the Soap port number to **8090**.

Change the Mms port to **101**.

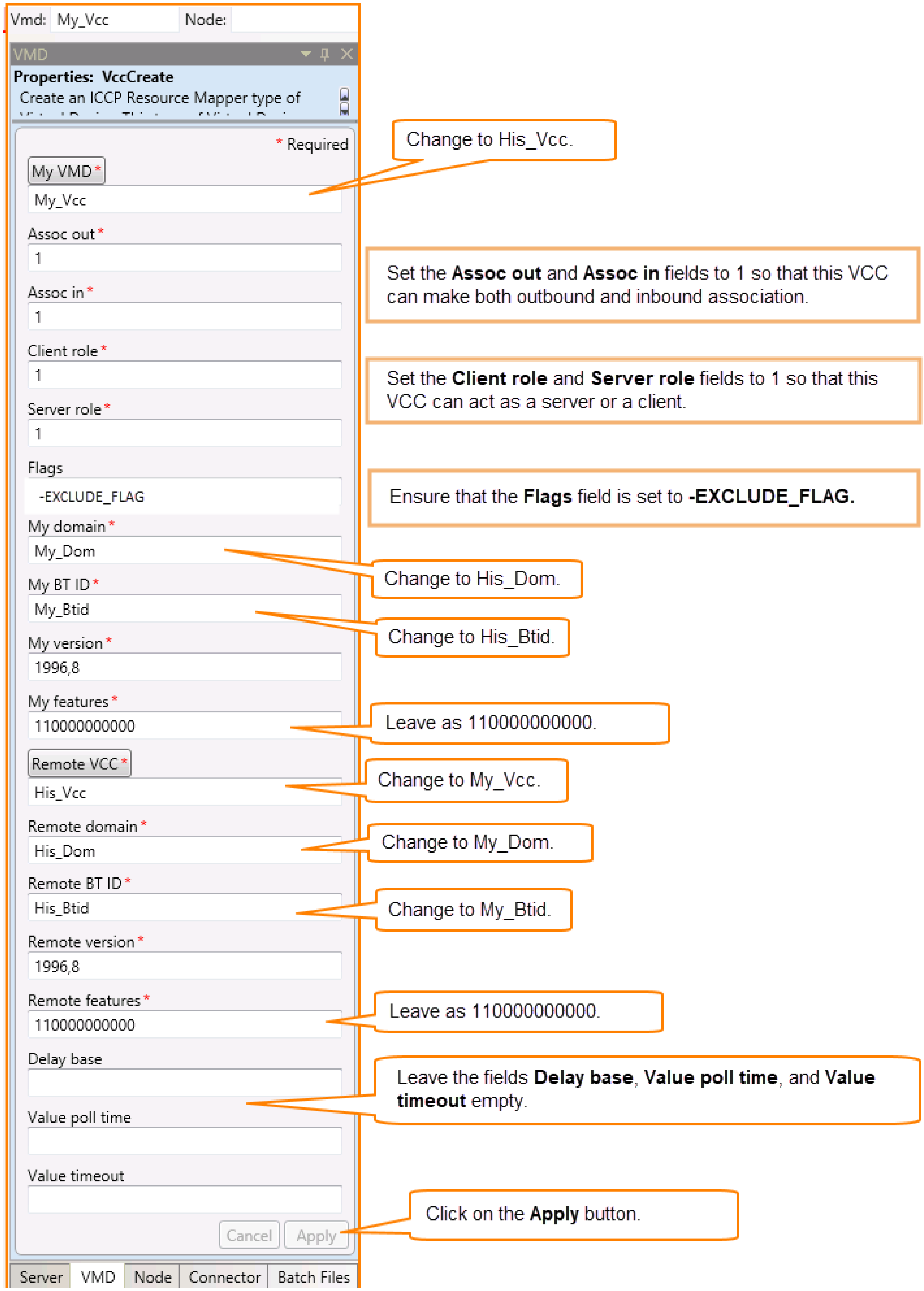
Click **Apply**.

Figure 73: Change Properties in the Server Panel

Change the My_Vcc VMD into His_Vcc

You need to change all references to My_Vcc to His_Vcc, and all references to His_Vcc to My_Vcc so that your local VMD becomes His_Vcc, and your remote Vcc becomes My_Vcc. You will also have to change My_Dom to His_Dom, and His_Dom to My_Dom; finally, you will have to change My_Btid to His_Btid, and His_Btid to My_Btid as shown in Figure 74.

1. Click on My_Vcc and change its label to His_Vcc.
2. Then make all of the changes specified in Figure 74.
3. Click the **Apply** button when you have finished making these changes.



Vmd: My_Vcc Node:

VMD

Properties: VccCreate

Create an ICCP Resource Mapper type of

* Required

My VMD *	My_Vcc
Assoc out *	1
Assoc in *	1
Client role *	1
Server role *	1
Flags	-EXCLUDE_FLAG
My domain *	My_Dom
My BT ID *	My_Btid
My version *	1996,8
My features *	110000000000
Remote VCC *	His_Vcc
Remote domain *	His_Dom
Remote BT ID *	His_Btid
Remote version *	1996,8
Remote features *	110000000000
Delay base	
Value poll time	
Value timeout	

Change to His_Vcc.

Set the **Assoc out** and **Assoc in** fields to 1 so that this VCC can make both outbound and inbound association.

Set the **Client role** and **Server role** fields to 1 so that this VCC can act as a server or a client.

Ensure that the **Flags** field is set to **-EXCLUDE_FLAG.**

Change to His_Dom.

Change to His_Btid.

Leave as 110000000000.

Change to My_Vcc.

Change to My_Dom.

Change to My_Btid.

Leave as 110000000000.

Leave the fields **Delay base**, **Value poll time**, and **Value timeout** empty.

Click on the **Apply** button.

Figure 74: Change My_Vcc, His_Vcc, My_Dom, His_Dom, My_Btid, and His_Btid

Change #client_real to #server_real and #client_state to #server_state in His_Vcc

Now you will need to change #client_real into #server_real so the following takes place:

- #client_real is fed data generated by the script Volts.py.
- #server_real is fed data communicated from My_Vcc.
- #client_state is fed data generated by the script Volts.py.
- #server_state is fed data communicated from My_Vcc.

You will have follow these steps:

1. Select His_Vcc.
2. Select #client_real.
3. Change the Point label to #client_real1.
4. Click the **Apply** button.
5. Select #client_state.
6. Change the Point label to #client_state1.
7. Click the **Apply** button.
8. Select #server_real.
9. Change the Point label to #client_real.
10. Change the Combine group to CliReals.
11. Click the **Apply** button.
12. Select #server_state.
13. Change the Point label to #client_state.
14. Change the Combine group to CliStates.
15. Click the **Apply** button.
16. Select #client_real1.
17. Change the Point label to #server_real.
18. Change the Combine group to SrvReals.
19. Click the **Apply** button.
20. Select #client_state1.
21. Change the Point label to #server_state.
22. Change the Combine group to #SrvStates.
23. Click the **Apply** button.

Change #client_real to #server_real and #client_state to #server_state in Processor

1. Select the Processor VMD.
2. Select #client_real.
3. Change the Point label to #client_real1.
4. Click the **Apply** button.
5. Select #client_state.
6. Change the Point label to #client_state1.
7. Click the **Apply** button.
8. Select #server_real.
9. Change the Point label to #client_real.
10. Change the Combine group to CliReals.
11. Click the **Apply** button.
12. Select #server_state.

13. Change the Point label to #client_state.
14. Change the Combine group to CliStates.
15. Click the **Apply** button.
16. Select #client_real1.
17. Change the Point label to #server_real.
18. Change the Combine group to SrvReals.
19. Click the **Apply** button.
20. Select #client_state1.
21. Change the Point label to #server_state.
22. Change the Combine group to #SrvStates.
23. Click the **Apply** button.

Change #client_real to #server_real and #client_state to #server_state in Script

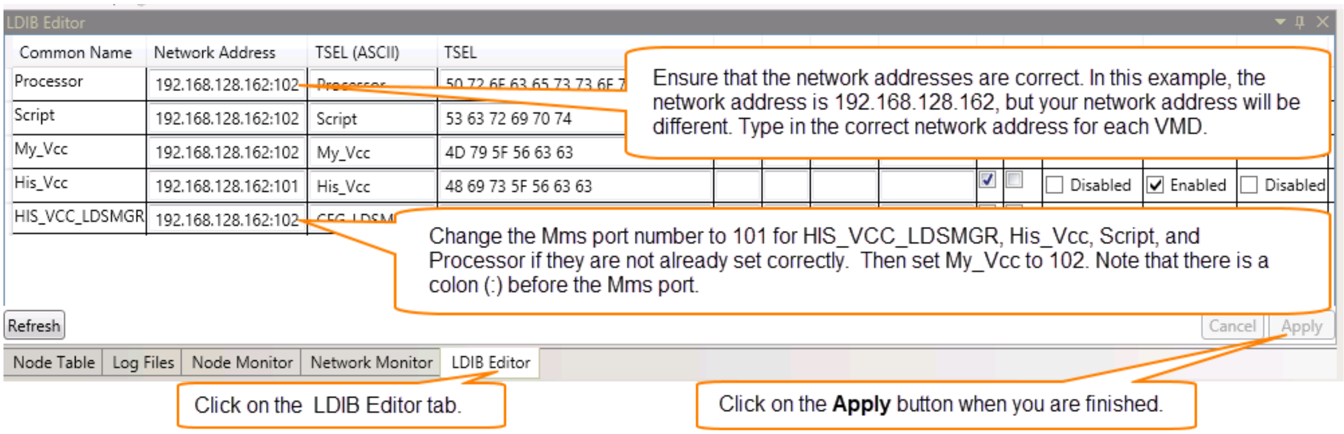
Now you will need to change #client_real into #server_real in the Script VMD:

You will have to follow these steps:

1. Select the Script VMD.
2. Select #client_real.
3. Change the Point label to #client_real1.
4. Click the **Apply** button.
5. Select #client_state.
6. Change the Point label to #client_state1.
7. Click the **Apply** button.
8. Select #server_real.
9. Change the Point label to #client_real.
10. Change the Combine group to CliReals.
11. Click the **Apply** button.
12. Select #server_state.
13. Change the Point label to #client_state.
14. Change the Combine group to CliStates.
15. Click the **Apply** button.
16. Select #client_real1.
17. Change the Point label to #server_real.
18. Change the Combine group to SrvReals.
19. Click the **Apply** button.
20. Select #client_state1.
21. Change the Point label to #server_state.
22. Change the Combine group to #SrvStates.
23. Click the **Apply** button.

Check Network Addresses in the LDIB Editor

1. Click on the LDIB Editor tab beneath the Node Table. If nothing appears, click the **Refresh** button in the bottom left corner of the panel.



Common Name	Network Address	TSEL (ASCII)	TSEL
Processor	192.168.128.162:102	Processor	50 72 6F 63 65 73 73 6F 7
Script	192.168.128.162:102	Script	53 63 72 69 70 74
My_Vcc	192.168.128.162:102	My_Vcc	4D 79 5F 56 63 63
His_Vcc	192.168.128.162:101	His_Vcc	48 69 73 5F 56 63 63
HIS_VCC_LDSMGR	192.168.128.162:102	CFG_LDSM	

Ensure that the network addresses are correct. In this example, the network address is 192.168.128.162, but your network address will be different. Type in the correct network address for each VMD.

Change the Mms port number to 101 for HIS_VCC_LDSMGR, His_Vcc, Script, and Processor if they are not already set correctly. Then set My_Vcc to 102. Note that there is a colon (:) before the Mms port.

Click on the LDIB Editor tab.

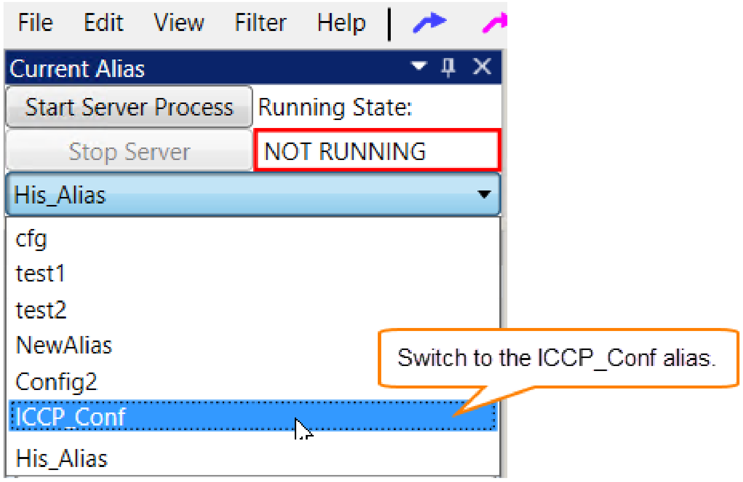
Click on the **Apply** button when you are finished.

2. The following table shows the MMS port for each VMD, the TSEL (ASCII) string , and the TSEL (hexadecimal) representation.

Common Name	MMS port	TSEL (ASCII)	TSEL (Hexadecimal)
HIS_ALIAS_LDSMGR	101	HIS_ALIAS_ LDSMGR	48 49 53 5f 41 4c 49 41 53 5f 4c 44 53 4d 47 52
Processor	101	Processor	50 72 6F 63 65 73 73 6F 72
My_Vcc	102	My_Vcc	4D 79 5F 56 63 63
Script	101	Script	53 63 72 69 70 74
His_Vcc	101	His_Vcc	48 69 73 5F 56 63 63

3. Click the **Apply** key to apply your changes.

4. Change back to the ICCP_Conf alias.



Switch to the ICCP_Conf alias.

5. Click on the LDIB tab beneath the Node Table panel.

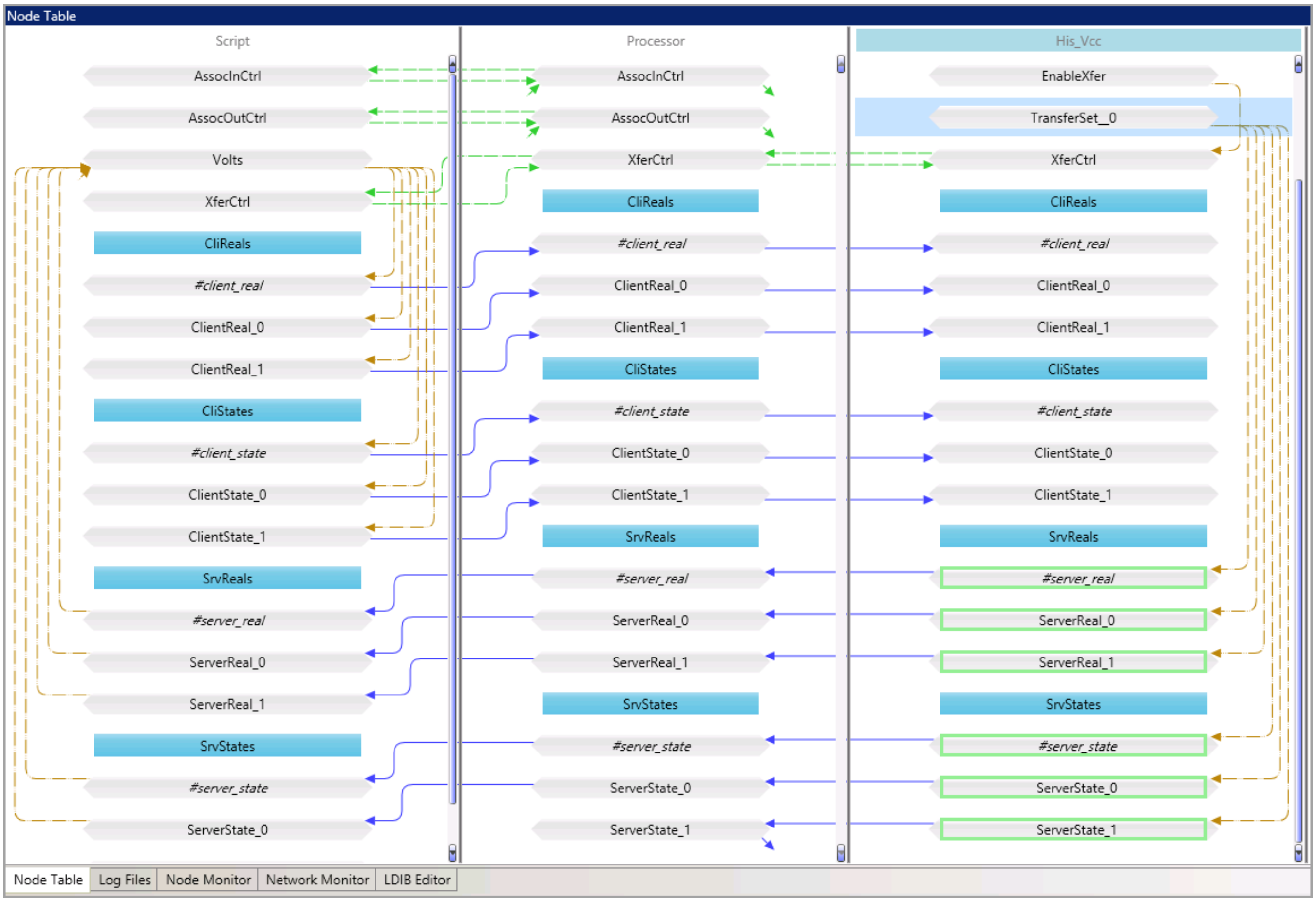
6. The following table shows the MMS port for each VMD, the TSEL (ASCII) string , and the TSEL (hexadecimal) representation.

Common Name	MMS port	TSEL (ASCII)	TSEL (Hexadecimal)
ICCP_CONF_LDSMGR	102	ICCP_CONF_ LDSMGR	49 43 43 50 5F 43 4F 4E 46 5F 4C 44 53 4D 47 52
Processor	102	Processor	50 72 6F 63 65 73 73 6F 72
My_Vcc	102	My_Vcc	4D 79 5F 56 63 63
Script	102	Script	53 63 72 69 70 74
His_Vcc	101	His_Vcc	48 69 73 5F 56 63 63

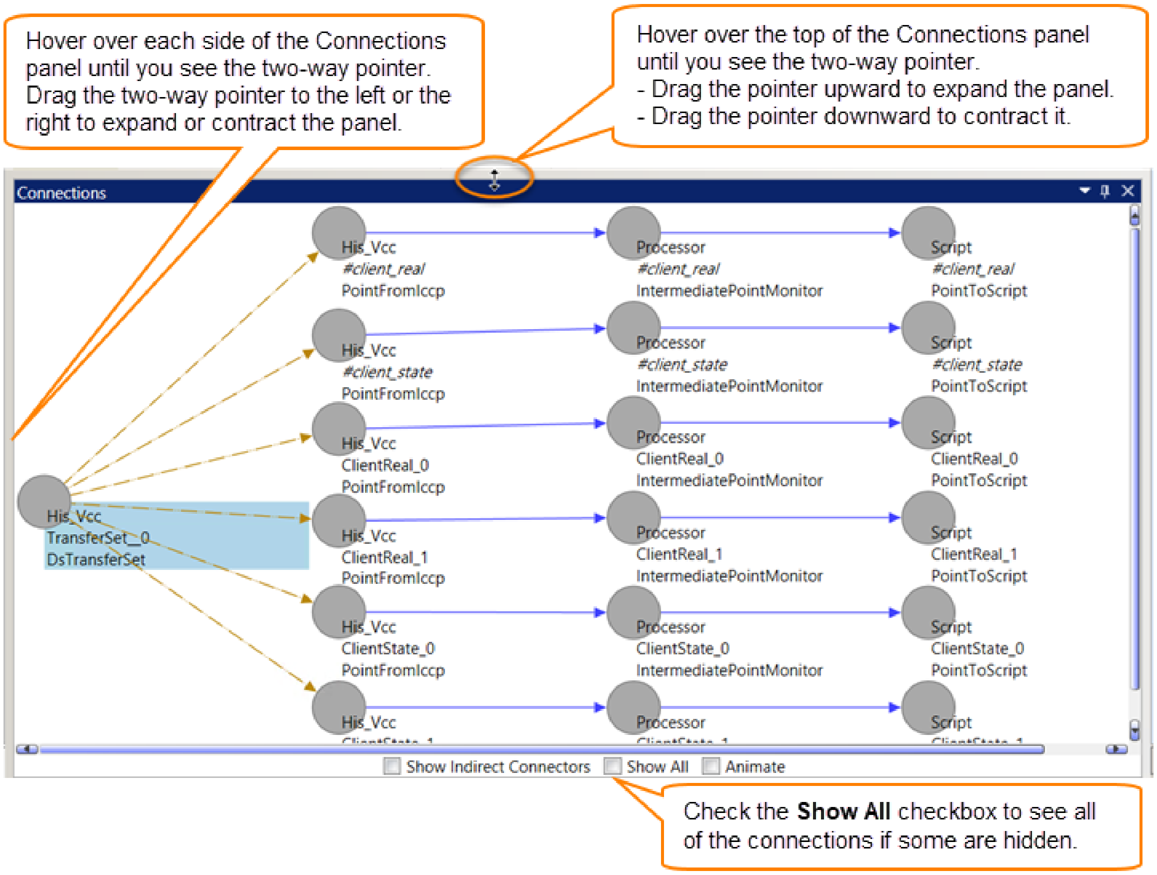
7. Click the **Apply** key to apply your changes.

Load a Batch File

1. Return to the His_Vcc alias.
2. Click on the Batch Files tab beneath the Properties panel.
3. Click on the **Load No Share** button.
4. Navigate to LiveData>Config.
5. Select IccpScriptTemplate.
6. Click on the **Open** button.



You can inspect all node connections from any node by clicking on the node in the Node Table then looking down at the Connections panel beneath the Node Table. For example, select the TransferSet_0 node in the His_Vcc VMD, and you will see all the connections that TransferSet_0 has.



Hover over each side of the Connections panel until you see the two-way pointer. Drag the two-way pointer to the left or the right to expand or contract the panel.

Hover over the top of the Connections panel until you see the two-way pointer.
- Drag the pointer upward to expand the panel.
- Drag the pointer downward to contract it.

Check the **Show All** checkbox to see all of the connections if some are hidden.

To Start the Server Processes

1. Click on the **Start Server Process** button directly underneath the Current Alias header to start the His_Vcc configuration.

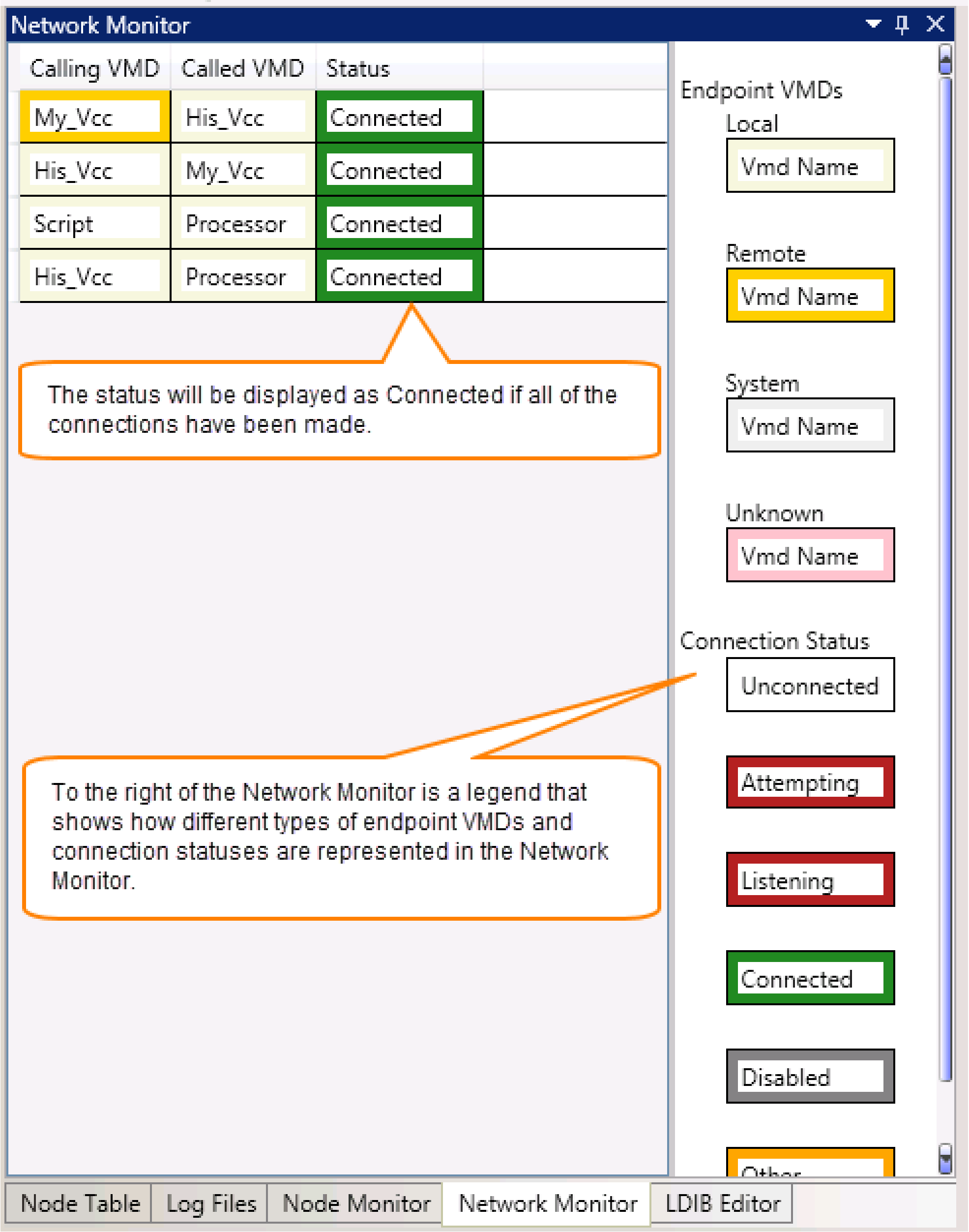
2. Return to the alias ICCP_Conf. Then load the batch file:

   - Click on the Batch Files tab beneath the Properties panel.
   - Click on the **Load No Share** button.
   - Navigate to LiveData>Config.
   - Select IccpScriptTemplate.
   - Click on the **Open** button to load the batch file.

3. Click on the Services tab, and start ICCP_Conf from this panel.

To Check that Associations Have Been Made Between All the VCCs

After you have started both configurations, return to the His_Vcc alias, and click on the Network Monitor tab at the bottom.



Calling VMD	Called VMD	Status
My_Vcc	His_Vcc	Connected
His_Vcc	My_Vcc	Connected
Script	Processor	Connected
His_Vcc	Processor	Connected

The status will be displayed as Connected if all of the connections have been made.

To the right of the Network Monitor is a legend that shows how different types of endpoint VMDs and connection statuses are represented in the Network Monitor.

ICCP Reference

One ICCP VMD and 20 ICCP node templates are provided for you in RTI Configuration Manager. This section provides an alphabetized reference of these templates. RTI Server provides support for many of the conformance blocks of the ICCP protocol.

Four of the node templates are designed to establish incoming and outgoing associations with other VMDs:

- VccAssocInControl (Setup Node)
- VccAssocOutControl (Setup Node)
- VccTransferControl (Setup Node)
- VerifyAssociation (Setup Node)

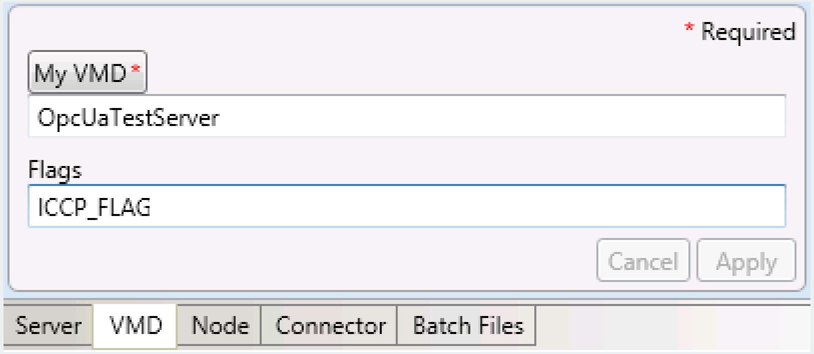
These node templates are common to all ICCP communication and have no specific block designation.

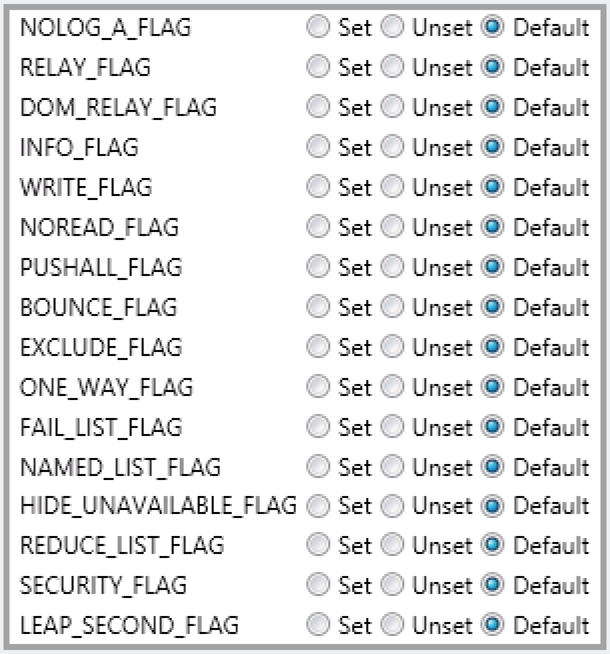
The following table describes the supported conformance blocks and references the templates associated with each block.

Block	Definition	Related Templates
Block 1 **Periodic System Data**	Includes status points, analogue points, quality flags, time stamp, change of value counter, protection events, and association objects to control ICCP sessions.	DsTransferSet (Setup Node for Blocks 1 and 2) PointFromIccp (Input Node for Blocks 1 and 2) PointToIccp (Output Node for Blocks 1 and 2)
Block 2 **Extended Data Set Condition Monitoring**	Provides report by exception capability for the data types that block 1 is able to transfer periodically.	
Block 4 **Information Messages**	Provides support for simple text and binary files.	ImTransferSet (Setup Node for Block 4) MessageFromIccp (Input Node for Block 4) MessageToIccp (Output Node for Block 4)
Block 5 **Device Control**	Provides device control requests: on/off, trip/close, raise/lower and digital set points. Includes mechanisms for interlocked controls and select-beforeoperate.	ControlFromIccp (Input Node for Block 5) ControlToIccp (Output Node for Block 5)
Block 8 **Additional User Objects**	Provides scheduling, accounting, and outage and plant information. The term "scheduling" refers to scheduling an amount of electrical power to be transferred from one system to another on a periodic basis for a certain interval of time under the restrictions of a formal agreement. From a data exchange standpoint, "scheduling" is expanded to include the retrieval of any periodic or profile data for control center energy scheduling, accounting, or application monitoring.	GetOutageFromIccp (Input Node for Block 8) GetTAQueryFromIccp (Input Node for Block 8) SendOutageToIccp (Output Node for Block 8) SendTAQueryToIccp (Output Node for Block 8) TANoSegPeriodicFromIccp (Input Node for Block 8) TANoSegPeriodicToIccp (Output Node for Block 8) TAServerTransferSetFromIccp (Input Node for Block 8) TaTransferSet (Setup Node for Block 8)

VccCreate (VMD)

Creates an ICCP Resource Mapper type of Virtual Manufacturing Device (VMD). This type of Virtual Device communicates to other Devices via the ICCP protocol, also known as TASE.2. The VMD created by this macro is also called a VCC (Virtual Control Center).

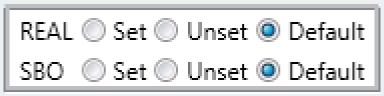
Parameter	Definition	Default
My VMD	Required. Specifies the common name to be assigned to this VCC.	No default
Assoc out	Required. Specifies whether this VCC is to have an outbound association. 1 enables an outbound association; 0 does not.	No default
Assoc in	Required. Specifies whether this VCC is to have an inbound association. 1 enables an inbound association; 0 does not. Note: A VCC can have both and inbound and outbound association.	No default
Client role	Required. Specifies whether the VCC has the ICCP client role or not. 1 specifies the client role; 0 disables it.	No default
Server role	Required. Specifies whether the VCC has the ICCP server role or not. 1 specifies the server role; 0 disables it.	No default
Flags	Optional. This parameter is a set of boolean (on/off) options that enable or disable certain functions in the ICCP VMD (VCC) that you are creating. The flags argument is expressed as a series of flag keywords separated by the vertical bar (' | ') character in the Flags field.  My VMD: OpcUaTestServer Flags: ICCP_FLAG * Required My VMD * Cancel Apply Server VMD Node Connector Batch Files	The PUSHALL_FLAG, EXCLUDE_FLAG, FAIL_LIST_FLAG, and NAMED_LIST_FLAG flags are set by default. All other flags are not set by default unless both the Assoc out and Assoc in parameters are set to ' 1 ', then the ONE_WAY_FLAG is set by default.

Parameter	Definition	Default
Flags	If you prefer to use a form, then click in the Flags field to view the form that allows you to set and unset flags using radio buttons.  NOLOG_A_FLAG ○ Set ○ Unset ● Default RELAY_FLAG ○ Set ○ Unset ● Default DOM_RELAY_FLAG ○ Set ○ Unset ● Default INFO_FLAG ○ Set ○ Unset ● Default WRITE_FLAG ○ Set ○ Unset ● Default NOREAD_FLAG ○ Set ○ Unset ● Default PUSHALL_FLAG ○ Set ○ Unset ● Default BOUNCE_FLAG ○ Set ○ Unset ● Default EXCLUDE_FLAG ○ Set ○ Unset ● Default ONE_WAY_FLAG ○ Set ○ Unset ● Default FAIL_LIST_FLAG ○ Set ○ Unset ● Default NAMED_LIST_FLAG ○ Set ○ Unset ● Default HIDE_UNAVAILABLE_FLAG ○ Set ○ Unset ● Default REDUCE_LIST_FLAG ○ Set ○ Unset ● Default SECURITY_FLAG ○ Set ○ Unset ● Default LEAP_SECOND_FLAG ○ Set ○ Unset ● Default	The EXCLUDE_FLAG is set by default.
	To remove a flag, unset it in the form or precede it with a minus sign in the Flags field.	
	NOLOG_A_FLAG: If set, directs RTI Server not to generate log messages for variable access failures.	Not Set
	RELAY_FLAG: If set, directs RTI Server to treat incoming Information Reports and Read Event Notifications as though they were MMS Writes to a local variable by the same name as specified in the report.	Not Set
	DOM_RELAY_FLAG: If set, directs RTI Server to treat incoming Information Reports and Read Event Notifications as though they are Writes, creating a domain-scoped variable where the domain is the remote VMD name.	Not Set
	INFO_FLAG: If set, directs RTI Server to send out Information Reports instead of Event Notifications for pushed and polled variables and for enrolled MMS Read Events.	Not Set
	WRITE_FLAG: If set, directs RTI Server to send out MMS Writes instead of Event Notifications for pushed and polled variables and for enrolled MMS Read Events.	Not Set
	NOREAD_FLAG - If set, directs RTI Server not to perform an initial read of all push-list variables, directing RTI Server to refresh the variable's internal state only as writes come in.	Not Set
	PUSHALL_FLAG: If set , directs RTI Server not to check push-list variables for changes in value, directing RTI Server to push a variable whenever a write comes in regardless of the value.	Not Set
	BOUNCE_FLAG: If set, directs RTI Server to allow a write to a variable on a given association to trigger a notification (push list or push agent) on the same association.	Not Set

Parameter	Definition	Default
	EXCLUDE_FLAG - If set, directs RTI Server not to allow incoming connections to this VMD except for those whose names appear in an 'AllowMmsInbound' node.	Set
	ONE_WAY_FLAG: If set, directs RTI Server to force outbound requests to use an outbound association.	Not Set
	FAIL_LIST_FLAG: If set, directs RTI Server to prevent the creation of a variable list if any of its variables are non-existent.	Not Set
	NAMED_LIST_FLAG: If set, directs RTI Server to perform an integrity scan with RBE to send a named variable list instead of an enumerated list.	Not Set
	HIDE_UNAVAILABLE_FLAG: If set, directs RTI Server to treat any variable whose value cannot be read as though the variable does not exist.	Not Set
	REDUCE_LIST_FLAG - If set and the remote ICCP server refuses to define a data set, causes the ICCP Client to attempt to read the data values and then retry the data set definition with any inaccessible data values omitted.	Not Set
	SECURITY_FLAG - If set, directs RTI Server to perform the secure ICCP procedures (certificate checking) at the ACSE level.	Not Set
	LEAP_SECOND_FLAG: If set, directs RTI Server to adjust timestamps according to the official schedule of leap seconds (as found in the LiveData.ini file).	Not Set
My domain	Required. Specifies the local domain name for server-side data.	No default
My BT ID	Required. Specifies the local bilateral table ID.	No default
My version	Required. Specifies the local ICCP version, such as 1996,8.	No default
My features	Required. Specifies locally supported features, as twelve ones or zeros or a combination of ones and zeroes.	No default
Remote VCC	Required. Specifies the common name of the remote VCC. Use the name of the server VCC if the client and server are different.	No default
Remote domain	Required. Specifies the remote domain name for server-side data.	No default
Remote BT ID	Required. Specifies the remote bilateral table ID.	No default
Remote version	Required. Specifies the remote ICCP version, such as 1996,8.	No default
Remote features	Required. Specifies the features which must be supported by remote VCC, as twelve ones or zeros, or a combination of ones and zeros.	No default
Delay base	Optional. Specifies the reference time in seconds relative to the present time for StartDelay (in the DsTransferSet macro).	No default
Value poll time	Optional. Specifies the period in seconds for polling ICCP server variables to test for the availability of their values.	5 seconds
Value timeout	Optional. Specifies the seconds that RTI Server is to wait for all variables to have a value before allowing an association. A value of 0 disables the feature.	0 seconds

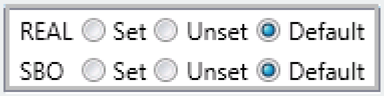
ControlFromIccp (Input Node for Block 5)

Takes device control from the ICCP peer.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager, for example, when connecting this node to another node.	Yes
My domain	Specifies the local domain name for server-side data. Leave this parameter blank if the scope is VCC-wide.	Yes, unless the scope is VCC-wide.
Flags	This parameter is a set of boolean (on/off) options that enable or disable certain functions in the OPC UA VMD that you are creating. The flags argument is expressed as a series of flag keywords separated by the vertical bar (' | ') character in the Flags field.  REAL ○ Set ○ Unset ● Default SBO ○ Set ○ Unset ● Default If you prefer to use a form, then click in the Flags field to view the form that allows you to set and unset flags using radio buttons. To remove a flag, unset it in the form or precede it with a minus sign in the Flags field.	No, if the data type is COMMAND.
	REAL: Sets the control point to type REAL. Default: If not set, the default is type COMMAND.	
	DISCRETE: Sets the control point to type DISCRETE. Default: If not set, the default is type COMMAND.	
	SBO: Indicates whether the control point supports the select-before-operate sequence.	Yes, if you are using SBO.
	TAGABLE: Indicates whether the control point supports tagging.	No
Combine group	This is the name used for grouping nodes in this VMD when they are displayed in the Node Table. Specify a combine group name if you would like to group nodes together when they serve a similar function.	No

ControlToIccp (Output Node for Block 5)

Transfers device control to the ICCP peer.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager, for example, when connecting this node to another node.	Yes
Remote VCC	Specifies the common name of the remote VCC. Use the name of the server VCC if the client and server are different.	Yes
Remote domain	Specifies the remote domain name for server-side data. Leave this parameter blank if the scope is VCC-wide.	Yes, unless the scope is VCC-wide.
Flags	This parameter is a set of boolean (on/off) options that allows you to select a data type and specify additional options. The flags argument is expressed as a series of flag keywords separated by the vertical bar (' | ') character in the Flags field.  REAL ○ Set ○ Unset ● Default SBO ○ Set ○ Unset ● Default If you prefer to use a form, then click in the Flags field to view the form that allows you to set and unset flags using radio buttons. To remove a flag, unset it in the form or precede it with a minus sign in the Flags field.	No, if the data type is COMMAND.
	REAL: Sets the control point to type REAL. Default: If not set, the default is type COMMAND.	
	DISCRETE: Sets the control point to type DISCRETE. Default: If not set, the default is type COMMAND.	
	SBO: Indicates whether the control point supports the select-before-operate sequence.	Yes, if you are using SBO.
	TAGABLE: Indicates whether the control point supports tagging.	No
Combine group	This is the name used for grouping nodes in this VMD when they are displayed in the Node Table. Specify a combine group name if you would like to group nodes together when they serve a similar function.	No

DsTransferSet (Setup Node for Blocks 1 and 2)

Defines a client-side DS transfer set.

Parameter	Definition	Required
TS num	Assigns a transfer set number that is unique to this VCC.	Yes
Remote domain	Specifies the remote domain for data.	Yes
Remote VCC	Specifies the common name of the remote server's VCC.	Yes
StartTime	Specifies the absolute start time as GMTBasedS.	No
StartDelay	Specifies the start time in seconds relative to the #delay_base in the VccCreate VMD.	No
IntervalCheck	Specifies the period in seconds between interval-based transmissions.	Yes
TLE	Specifies the time limit for execution in seconds.	No
BufferTime	Specifies the buffer time in seconds.	No
IntegrityCheck	Specifies the period in seconds between integrity-based transmissions.	Yes
IntervalTimeOut	Indicates whether interval based transmissions are enabled. 1 indicates that interval-based transmissions are enabled; 0 indicates that they are not.	Yes
IntegrityTimeOut	Indicates whether integrity-based transmissions are enabled. 1 indicates that integrity-based transmissions are enabled; 0 indicates that they are not.	Yes
ObjectChange	Indicates whether change-based transmissions are enabled. 1 indicates that change-based transmissions are enabled; 0 indicates that they are not.	Yes
OperatorRequest	Indicates whether operator-requested transmissions are enabled. 1 indicates that operator-requested transmissions are enabled; 0 indicates that they are not.	Yes
RBE	Indicates whether report-by-exception is enabled. 1 indicates that report-by-exception is enabled; 0 indicates that report-by-exception is not.	Yes
IntervalDelay	Specifies the start time delay after the next time, which is a multiple of the **IntervalCheck** period (e.g., if IntervalCheck is 3600 seconds, and IntervalDelay is 60 seconds, poll 1 minute after each hour). Defining this parameter (even with a 0 value) enables this feature.	No
AllChangesReported	Indicates whether or not all changes are to be reported. 1 indicates that the all-changes-reported function is enabled; 0 indicates that this function is not enabled.	No
Critical	Indicates whether critical acknowledgment is required or not. 1 indicates that critical acknowledgment is required; 0 or no value indicates that it is not.	No
CircumventSiemensBug	Indicates whether or not to circumvent the Siemens Transfer Set domain name bug. Specifying any value directs RTI Server is to circumvent the Siemens bug. No value indicates not to circumvent this bug.	No
Do read	Indicates whether or not RTI Server is to perform an initial read. 1 indicates to do an initial read of data values; 0 or no value indicates that RTI Server is not to perform an initial read.	No
Combine group	This is the name used for grouping nodes in this VMD when they are displayed in the Node Table. Specify a combine group name if you would like to group nodes together when they serve a similar function.	No

GetOutageFromIccp (Input Node for Block 8)

Retrieves outage data from the ICCP peer. Specifically, this outage data provides the time period when the outage occurred.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager. This specific label indicates the information requested about the outage. Specify one of the following: » DONewRevSched returns the date that the outage started and when it ended. » DOActual gets the date that the outage occurred. » DOCancel gets the cancellation date. The cancellation date is the time when the outage was cancelled if it was cancelled.	Yes
Type	Specifies the data type for this outage object. Use the data type with the same name as the Point label: » DONewRevSched » DOActual » DOCancel	Yes
Remote VCC	Specifies the common name of the remote VCC. Use the server VCC name if the client and server are different.	Yes
Remote domain	Specifies the remote domain for data. Leave this parameter blank if the scope of the data is VCC-wide.	Yes, unless the scope is VCC-wide.
Combine group	This is the name used for grouping nodes in this VMD when they are displayed in the Node Table. Specify a combine group name if you would like to group nodes together when they serve a similar function.	No

GetTAQueryFromIccp (Input Node for Block 8)

Retrieves the Transfer Account Query from the ICCP peer.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager, for example, when connecting this node to another node.	Yes
Remote VCC	Specifies the common name of the remote server VCC.	Yes

ImTransferSet (Setup Node for Block 4)

Defines a client-side information message (IM) transfer set.

Parameter	Definition	Required
TS num	Assigns a transfer set number that is unique to this VCC.	Yes
Remote VCC	Specifies the common name of the remote server's VCC.	Yes
Combine group	This is the name used for grouping nodes in this VMD when they are displayed in the Node Table. Specify a combine group name if you would like to group nodes together when they serve a similar function.	Yes

MessageFromIccp (Input Node for Block 4)

Receives an information message from the ICCP peer as an IM_struct type. It is recommended that you set the RELAY_FLAG for the ICCP association. Use an update connector for this point.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager, for example, when connecting this node to another node.	Yes
Remote VCC	Specifies the common name of the remote VCC. Use the server VCC if the client and server are different.	Yes
TS num	Specifies the number of the IM transfer set. The default is 0.	No
Size	Specifies the information buffer size in bytes. The default is 1024	No
Remote domain	Specifies the domain name used in incoming Block 4 information reports. Leave this parameter blank if the scope of the information reports is VCC-wide.	Yes, unless the scope is VCC-wide.

MessageToIccp (Output Node for Block 4)

Sends an information message to the ICCP Peer by writing an IM_struct type to this node. Use either an update or time-based connector.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager, for example, when connecting this node to another node.	Yes
Remote VCC	Specifies the common name of the remote client VCC.	Yes
My domain	Specifies the domain name that is used in outgoing Block 4 information reports. Leave this parameter blank if the scope of the information reports is VCC-wide.	Yes, unless the scope is VCC-wide.

PointFromIccp (Input Node for Blocks 1 and 2)

Retrieves indication point data from the ICCP peer.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager, for example, when connecting this node to another node.	Yes
Type	The supported ICCP types are: » Data_Discrete » Data_Real » Data_RealQ » Data_RealQTimeTag » Data_RealExtended » Data_RealQTimeTagExtended » Data_State » Data_StateQ » Data_StateQTimeTag » Data_StateExtended » Data_StateQTimeTagExtended » Data_Discrete » Data_DiscreteQ » Data_DiscreteQTimeTag » Data_DiscreteExtended » Data_DiscreteQTimeTagExtended	Yes
Remote VCC	Specifies the common name of the remote VCC. Use the server VCC if the client and server are different.	Yes
Remote domain	Specifies the remote domain for this point if the point is of domain scope. Leave blank for a VCC-wide point.	No
TS num	Specifies the number of the transfer set to which this point is assigned	No
Combine group	This is the name used for grouping nodes in this VMD when they are displayed in the Node Table. Specify a combine group name if you would like to group nodes together when they serve a similar function.	No
Period	Places a limit (in seconds) on the time RTI Server will wait between receiving reports before dis-establishing the association and then re-establishing it. Specify this limit on only one node associated with this VCC because specifying it on more than on requires more memory and will slow down performance. Note: The node must receive data at a specified time interval, not when the data changes; otherwise, the function that re-establishes the association will not work in the event that the data never changes. If this parameter is undefined or set to 0, RTI Server will wait indefinitely.	No
Variation	Indicates whether or not the RELAY_FLAG is for this ICCP association. Enter 1 if the RELAY_FLAG is set for the ICCP association; otherwise, enter 0 or leave blank.	No

PointToIccp (Output Node for Blocks 1 and 2)

Sends indication point data to the ICCP peer.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager, for example, when connecting this node to another node.	Yes
Type	The supported ICCP types are: » Data_Discrete » Data_Real » Data_RealQ » Data_RealQTimeTag » Data_RealExtended » Data_RealQTimeTagExtended » Data_State » Data_StateQ » Data_StateQTimeTag » Data_StateExtended » Data_StateQTimeTagExtended » Data_Discrete » Data_DiscreteQ » Data_DiscreteQTimeTag » Data_DiscreteExtended » Data_DiscreteQTimeTagExtended	Yes
My domain	Specifies the local domain for this point if the point has domain scope. Leave this parameter blank if the point's scope is VCC-wide.	Yes, unless the scope is VCC-wide.
Combine group	This is the name used for grouping nodes in this VMD when they are displayed in the Node Table. Specify a combine group name if you would like to group nodes together when they serve a similar function.	No

SendOutageToIccp (Output Node for Block 8)

Provides device outage data to the ICCP peer in order to execute a planned outage.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager. This specific label indicates information about the planned outage that RTI Server is to send to the ICCP peer. Specify one of the following: » DONewRevSched sends the dates when the outage is to start and end. » DOActual sends the date when the outage is to occur. » DOCancel sends the cancellation date if you or your organization decides to cancel the outage.	Yes
Type	Specifies the data type for this outage object. Use the data type with the same name as the Point label: » DONewRevSched » DOActual » DOCancel	Yes
Remote VCC	Specifies the X.500 common name of the remote client VCC.	Yes
My domain	Specifies the local VCC domain issuing this report. Leave this parameter blank if this node's scope is VCC-wide.	Yes, unless the scope is VCC-wide.
Combine group	This is the name used for grouping nodes in this VMD when they are displayed in the Node Table. Specify a combine group name if you would like to group nodes together when they serve a similar function.	No

SendTAQueryToIccp (Output Node for Block 8)

Sends a query to the ICCP peer.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager, for example, when connecting this node to another node.	Yes
Remote VCC	Specifies the common name of the remote server VCC.	Yes

TANoSegPeriodicFromIccp (Input Node for Block 8)

Retrieves "scheduling" and "accounting" information from the ICCP peer periodically. In particular, this type of node retrieves the information without segmentation.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager, for example, when connecting this node to another node.	Yes
Row type	Provides the MMS-DL (Manufacturing Messaging Service-Descriptive Language) specification of the type of data in a single matrix row.	Yes
Num floats	Specifies the number of floats in a matrix row.	Yes
Num ints	Specifies the number of integers in a matrix row.	Yes
Num matrix ids	Specifies the sum of the number of floats and integers in a matrix row.	Yes
Max loc refs	Specifies the maximum number of local refs that are allowed in the header structure	Yes
Max rows	Specifies the maximum number of matrix rows which may be passed using this point	Yes
Ts num	Specifies the number of the transfer set to which this point is assigned.	Yes
Combine group	This is the name used for grouping nodes in this VMD when they are displayed in the Node Table. Specify a combine group name if you would like to group nodes together when they serve a similar function.	No

TANoSegPeriodicToIccp (Output Node for Block 8)

Sends "scheduling" and "accounting" information to the ICCP peer periodically. In particular this type of node sends the information without segmentation.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager, for example, when connecting this node to another node.	Yes
Row type	Provides the MMS-DL specification of the type of a single matrix row.	Yes
Num matrix ids	Indicates how many numbers are in a matrix row. These numbers can be either integers or floats, or some combination of integers and floats.	Yes
Max loc refs	Specifies the maximum number of local refs that are allowed in the header structure.	Yes
Max rows	Specifies the maximum number of matrix rows which may be passed using this point.	Yes
My domain	Specifies the domain in the local VCC of this report. Leave this parameter blank if the scope is VCC-wide.	Yes, unless the scope is VCC-wide.

Combine group	This is the name used for grouping nodes in this VMD when they are displayed in the Node Table. Specify a combine group name if you would like to group nodes together when they serve a similar function.	No

TAServerTransferSetFromIccp (Input Node for Block 8)

Defines a server-side transfer account (TA) transfer set.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager, for example, when connecting this node to another node.	Yes
Remote VCC	Specifies the common name of the remote server VCC.	Yes

TaTransferSet (Setup Node for Block 8)

Defines a client-side transfer account (TA) transfer set.

Parameter	Definition	Default
Ts num	Assigns a transfer set number that is unique to this VCC.	Yes
Remote domain	Assigns a transfer set number that is unique to this VCC.	Yes
Remote VCC	Specifies the common name of the remote server VCC.	Yes
BeforeTheHour	Indicates whether or not the ICCP server is to send a report before the hour. Specify 1 to direct the ICCP server to send this information before the hour; otherwise, specify 0. Often, this is referred to as "pre-schedules ."	Yes
DispatchUpdate	Indicates whether or not the ICCP server is to send a report for a dispatch update. Specify 1 to direct the ICCP server to send a report for a dispatch update; otherwise, specify 0. Often, this is referred to as "next hour schedules . "	Yes
DuringTheHour	Indicates whether or not the ICCP server is to send a report during the hour. Specify 0.1 to direct the server to send these reports during the hour; otherwise, specify 0. Often, this is referred to as "mid hour schedule changes. "	Yes
AfterTheHour	Indicates whether or not the ICCP server is to send a report after the hour. Specify 1 to direct the server to send these a report after the hour; otherwise, specify 0. Often, this is referred to as "after the hour actuals."	Yes
ActualDataUpdate	Indicates whether or not the ICCP server is to send a report for an actual data update. Specify 1 to direct the server to send a report for an update; otherwise, specify 0. Often, this is referred to as "corrections to previous schedules."	Yes
PastHours	Indicates whether or not the ICCP server is to send a report for data in the past. Specify 1 to direct the server to send a report for this past data; otherwise, specify 0.	Yes
ObjectChange	Indicates whether or not the server is to send a report when any object in the transfer account changes. Specify 1 to direct the server to send a report when a change occurs; otherwise, specify 0.	Yes
OperatorRequest	Indicates whether or not the ICCP server is to send a report when an operator at the ICCP server control requests it. Specify 1 to direct the server to send a report upon operator request; otherwise, specify 0.	Yes

Note: A Transfer Account object represents what, where, when, and how much is transferred between two utilities in a particular account. It may also represent generation schedules and other energy delivery schedules within a utility. It is a container for a number of different attributes and objects, which together define the entire transfer account definition, i.e. which account, when is the effective time frame, and what are the periodic or profile values of the data.

VccAssocInControl (Setup Node)

Establishes inbound association control, enabling and disabling the inbound ICCP association to this VCC, and reporting the association status. This node must be linked to a node in another virtual device, usually the processor, with a TwoWayConnector. The data type is <integer:32>. Control values are -1 to abort and disable, +0 to disable, and +1 to listen for an association. Status values are +0 for disabled, +1 for listening for association, and +2 for associated.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager, for example, when connecting this node to another node.	Yes
Remote VCC	Specifies the common name of the remote VCC.	Yes
Init state var	Indicates whether or not RTI Server is to establish an association initially. Specify +1 if RTI Server is to enable the association initially; specify +0 if RTI Server is not to enable the association initially.	Yes
Association flags	Each of these flags disables or enables an option for handling data and associations. By default, none of these flags are set. The flags argument is expressed as a series of flag keywords separated by the vertical bar (' | ') character in the Flags field. If you prefer to use a form, then click in the Flags field to view the form that allows you to set and unset flags using radio buttons. To remove a flag, unset it in the form or precede it with a minus sign in the Flags field.	No
	RELAY_FLAG: If set, directs RTI Configuration Manager to treat incoming Information Reports and Read Event Notifications as though they were MMS Writes to a local variable by the same name as in the report.	
	DOM_RELAY_FLAG: If set, directs RTI Configuration Manager to treat incoming Information Reports and Read Event Notifications as though they are Writes, creating a domain-scoped variable where the domain is the remote VMD name.	
	INFO_FLAG: If set, directs RTI Configuration Manager to send out Information Reports instead of Event Notifications for pushed and polled variables and for enrolled MMS Read Events.	
	BOUNCE_FLAG - If set, directs RTI Configuration Manager not to allow a write to a variable on a given association to trigger a notification (push list or push agent) on the same association.	
	EN_OUT_FLAG - If set, directs RTI Configuration Manager to attempt to make an outbound association.	
	SUPERCEDE_FLAG - If set and an associate request comes in while there is already an association, directs RTI Configuration Manager to abort the old association and accept the new one. If not set, RTI Configuration Manager refuses the new association.	
	ALLOW_MULTI_FLAG - If set and an associate request comes in while there is already an association, directs RTI Configuration Manager to keep the old association and also accept the new one. If not set, refuse the new association and keep the old one.	
Combine group	This is the name used for grouping nodes in this VMD when they are displayed in the Node Table. Specify a combine group name if you would like to group nodes together when they serve a similar function.	No

VccAssocOutControl (Setup Node)

Provides outbound association control, enabling and disabling the outbound ICCP association from this VCC, and reporting the association status. This node must be linked to a node in another VMD (usually the processor) with a TwoWayConnector. The data type is <integer:32>. Control values are -1 to abort and disable, +0 to disable, and +1 to try to associate. Status values are +0 for disabled, +1 for trying to associate, and +2 for associated.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager, for example, when connecting this node to another node.	Yes
Remote VCC	Specifies the name of the remote VCC.	Yes
Init state var	Indicates whether or not RTI Server is to establish an association initially. Specify +1 if RTI Server is to enable the association initially; specify +0 if RTI Server is not to enable the association initially.	Yes

Association flags	Each of these flags disables or enables an option for handling associations controlled by this node. By default, none of these flags are set. The flags argument is expressed as a series of flag keywords separated by the vertical bar (' | ') character in the Flags field. If you prefer to use a form, then click in the Flags field to view the form that allows you to set and unset flags using radio buttons. To remove a flag, unset it in the form or precede it with a minus sign in the Flags field.	No
	RELAY_FLAG: If set, directs RTI Server to treat incoming Information Reports and Read Event Notifications as though they are MMS Writes to a local variable by the same name as in the report.	
	DOM_RELAY_FLAG: If set, directs RTI Server to treat incoming Information Reports and Read Event Notifications as though they are Writes, creating a domain-scoped variable where the domain is the remote VMD name.	
	INFO_FLAG: If set, directs RTI Server to send out Information Reports instead of Event Notifications for pushed and polled variables and for enrolled MMS Read Events.	
	WRITE_FLAG: If set, directs RTI Server to send out MMS Writes instead of Event Notifications for pushed and polled variables and for enrolled MMS Read Events.	
	NOREAD_FLAG: If set, directs RTI Server to make all AA-specific PGE variables candidates for pushing (without the need for inclusion in a PUSH list).	
	REPORT_FLAG: If set, directs RTI Server not to perform an initial read of all push-list variables, directing RTI Server to refresh the variable's internal state only as writes come in.	
	BOUNCE_FLAG: If set, directs RTI Server not to allow a write to a variable on a given association to trigger a notification (push list or push agent) on the same association.	
	EN_OUT_FLAG: If set, directs RTI Server to attempt to make an outbound association.	
	SUPERCEDE_FLAG: If set and an associate request comes in while there is already an association, directs RTI Configuration Manager to abort the old association and accept the new one. If not set, RTI Configuration Manager refuses the new association.	
	ALLOW_MULTI_FLAG: If set and an associate request comes in while there is already an association, directs RTI Server to keep the old association and also accept the new one. If not set, refuse the new association and keep the old one.	
Combine group	This is the name used for grouping nodes in this VMD when they are displayed in the Node Table. Specify a combine group name if you would like to group nodes together when they serve a similar function.	No

VccTransferControl (Setup Node)

Establishes transfer set control, enabling and disabling the client-side transfer sets (data set, message, and accounts) for this VCC, and reporting the transfer set status. This node must be linked to a node in another virtual device, usually the processor, with a TwoWayConnector.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager, for example, when connecting this node to another node.	Yes
Type	Specifies the data type of the transfer set, which must be a packed boolean array type, such as <boolean>(<pack>16).	Yes
Init state var	Specifies the name of a packed boolean array type variable, where the true (1) array elements initially enable the corresponding transfer sets, and the false (0) elements disable them.	Yes
Combine group	This is the name used for grouping nodes in this VMD when they are displayed in the Node Table. Specify a combine group name if you would like to group nodes together when they serve a similar function.	No

VerifyAssociation (Setup Node)

Periodically requests an ICCP Identify from the ICCP peer. If the request is not acknowledged by the time specified in the timeout parameter (default is 1 minute), then the association will be aborted. A new association can then be established.

Parameter	Definition	Required
Label	Assigns a name that is for routing the data within RTI Server. This name will be displayed throughout RTI Configuration Manager, for example, in the Node Table, and can be used in batch files in order to create more instances of this node. Use this label whenever you refer to the node within RTI Configuration Manager, for example, when connecting this node to another node.	Yes
Remote VCC	Specifies the common name of the remote VCC.	Yes
Pclass	Specifies the LiveData poll class controlling the frequency of issuing the identify request. 1 - specifies 10 seconds. 2 - specifies 1 seconds. 3 - specifies .1 seconds. Do not use poll class 0 because it is the PUSH class, which makes no sense in this case.	Yes
Timeout	Specifies the time-out in milliseconds on all confirmed requests for the entire server. The default is 60000 (one minute). This timeout will govern how soon the identify request (and all other requests) time out. **Note:** Because this parameter is global, specify this parameter only once.	No

Glossary

A

Alias

An alias is an internal configuration buffer. A configuration is imported from a file into this configuration buffer, named "cfg". While a configuration is in this buffer, RTI Configuration Manager can access it, allowing you to edit the configuration and execute the configuration on RTI Server. For testing purposes, there is a need for more than one active configuration so that you can test various improvements before incorporating them into the configuration. These testing areas are also in internal configuration buffers, usually called "test1" and "test2".

D

DNP

Distributed Network Protocol is a set of communications protocols used between components in process automation systems. Its main use is in utilities such as electric and water companies. Usage in other industries is not common. It was developed for communications between various types of data acquisition and control equipment. It plays a crucial role in SCADA systems, where it is used by SCADA Master Stations (a.k.a. Control Centers), Remote Terminal Units (RTUs), and Intelligent Electronic Devices (IEDs).

F

Filter Node

A filter node is used in a general sense to refer to a node through which data may pass within RTI Server as opposed to Input and Output nodes. Some filters modify numerical values passing through them; some change data types; some combine or split data, and some just act as intermediate reference points and do not modify data.

I

ICCP

The Inter-Control Center Communications Protocol (ICCP or IEC 60870-6/TASE.2)[1] is specified by utility organizations throughout the world to provide data exchange over wide area networks (WANs) between utility control centers, utilities, power pools, regional control centers, and Non-Utility Generators. ICCP is also an international standard: International Electrotechnical Commission (IEC) Telecontrol Application Service Element 2 (TASE.2).

Input Node

A node that brings information into LiveData RTI Server from the external world by way of one of the protocols supported by RTI Server. There are, however, some input nodes that provide access information within RTI Server, such as a constant, system variable, or Memory VMD.

O

Output Node

An output node generally sends information out from RTI Server to the external world by way of on of the protocols supported by RTI Server. There are, however, some output nodes that provide access to information within

RTI Server, such as nodes in a Memory VMD, which map onto RTI Server memory.

P

Point

1. A point is an independent datum that passes through an RTI configuration. For example, a single ICCP node might receive a value (datum) from the ICCP peer, then pass the value through multiple nodes (which might perform operations on the datum) in multiple VMDs, and finally output the value to another device, such as a DNP device. Although the point would travel from an input node to various other nodes and exit through an output node, in this definition of a point, this datum would refer to a single point. 2. A point represents a single input or output value, or an intermediate value, that is controlled by RTI Server. A hard point represents an actual input or output within the system, while a soft point results from logic and math operations applied to other points. Many of the points within a configuration are soft points, containing intermediate values. RTI Configuration Manager removes the distinction between hard and soft points.

R

RTU

Short for remote terminal unit. In SCADA systems, an RTU is a device installed at a remote location that collects data, codes the data into a format that is transmittable and transmits the data back to a central station, or master. An RTU also collects information from the master device and implements processes that are directed by the master. RTUs are equipped with input channels for sensing or metering, output channels for control, indication or alarms and a communications port.

S

Scattered variables

Variables that gather data points, which logically should be grouped together, but whose addresses are not in sequence. Once you define a scattered variable, LiveData treats it like a standard structure. Scattered variables can gather different data types from scattered locations in device memory, eliminating the need for control programs to copy and re-cast data. Scattered variables are designated as "vscatter" instead of "variable."

Setup Node

A Setup node can provide general information to RTI Server, such as poll classes and association lists; some Setup nodes are involved in controlling associations, and some Setup nodes provide for bulk transfer of input or output data, such as setting up a script or an ICCP transfer set to handle a group of points. In general, a Setup node does not deal with data on a point by point basis. Some Setup nodes have parameters containing a Common Name that refers to a block of network addressing information, which is known as LDIB.

V

VCC

VCC is short for Virtual Control Center, which is LiveData's term for an ICCP virtual device.

Index