

Oracle Utilities Live Energy Connect Configuration Manager User Guide

F48094-03

Last Updated October 23, 2023

Contents

Getting Started	5
Introduction to Live Energy Connect	5
Live Energy Connect Concepts and Terminology	6
ICCP and MMS	6
VMD	6
Configuration Aliases	7
Configuration Manager VMD	7
Points	8
Nodes	8
Connectors	9
Filter Nodes	9
Prototype Configurations	9
Batch File	10
PDI Macros	10
LEC User Interface	10
Current Alias Panel	11
Central Panel	12
Directly Connected Nodes Panel	13
Properties Panel	13
Connections Panel	16
Templates Panel	17
Menu Bar	18
A Quick Configuration Manager Tutorial	21
Scenario for the Example ICCP Client Configuration	21
Importing Oracle Utilities Live Energy Connect Server Configurations	22
Loading a Batch File	23
Starting and Stopping the Live Energy Connect Server	25
Creating a New Configuration Alias	26
Loading the Simulated SCADA System Configuration	26
Loading a Batch File for a Simulated SCADA Server Configuration	27
Adjusting the Server Parameters for the Simulated SCADA Server Configuration	27
Creating a Service for the Simulated SCADA Server Configuration	29
Adjusting the Network Information Using the LDIB Editor	29
Adjusting the Server Parameters for the Simulated SCADA Server Configuration	30

Additional Tools and Features in LEC Configuration Manager	32
Live Energy Connect Logging	32
Specifying the Visible Nodes in the Configuration Manager	34
Variable Access Tool	35
LDIB Editor	37
Gathering Diagnostic Information for Support	40
Running Live Energy Connect as a Process	40
Reloading Macros	41
Creating an ICCP Client	41
Step 1: Open an Empty Configuration	42
Step 2: Create a Processor VMD	42
Step 3: Create a VCC	44
Step 4: Create a Script VMD	50
Step 5: Arrange the VMD's in the Node Monitor	51
Step 6: Create Nodes in the VCC	52
Step 7: Create Nodes in the Processor VMD	60
Step 8: Create Nodes in the Script VMD	62
Step 9: Connect the Nodes	65
Step 10: Validate the Configuration	67
ICCP Reference	69
ICCP Indication Point Types in Oracle Utilities Live Energy Connect	69
ICCPQualityCodes	73
Oracle Utilities Live Energy Connect Configuration Manager Node Templates for ICCP	74

Oracle Utilities Live Energy Connect Configuration Manager User Guide

F48094-03

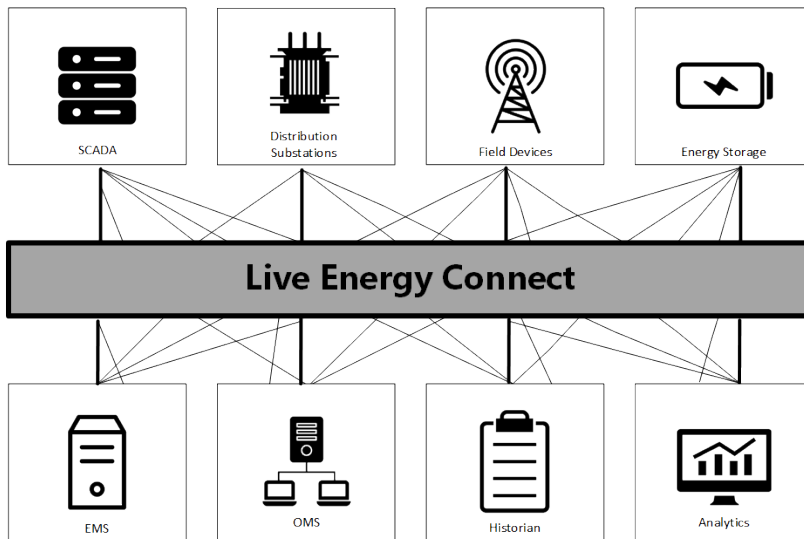
[Copyright ©](#)

Getting Started

The Oracle Utilities Live Energy Connect (LEC) Configuration Manager is a GUI application that gives users the ability to design, modify, and monitor their LEC Server configurations.

Introduction to Live Energy Connect

LEC is a highly configurable middleware product that enables real-time communication between any number of IT and OT systems.



LEC simplifies and reduces the interfaces between OT systems. In the above figure, the gray connectors represent the many possible interfaces between systems in a business without middleware. The solid black lines show how you can use LEC to reduce the complexity of your organization's systems.

Currently, LEC supports 20+ protocols used in the Utilities industry, including MMS, ICCP (TASE.2), DNP, Modbus, and OPC UA.

This document assumes the reader has access to a recent release of LEC (v7.0.0.0.0 or later). For information about installing LEC products, refer to the LEC Installation Guide or contact [My Oracle Support](#).

Note: LEC was formerly known as LiveData Utilities Real-Time Integration (RTI) Server Platform.

Live Energy Connect Concepts and Terminology

This section provides a list of concepts and terms used to describe Oracle Utilities Live Energy Connect (LEC) configurations. Some of these terms are specific to LEC, and some terms originate from industry protocols and standards.

On this page:

ICCP and MMS

The Inter-Control Center Communications Protocol (ICCP), also known as TASE.2, is a protocol that allows for real-time data exchange between utility systems over WANs or LANS. The ICCP standard is maintained by the International Electrotechnical Commission as IEC 60870.

For more information about ICCP, see [ICCP Reference](#).

ICCP itself is built on top of the Manufacturing Message Specification (MMS) standard, which is maintained by International Standards Organization as ISO 9506. MMS defines a standard for data exchange consisting of real-time monitoring and control data between two devices.

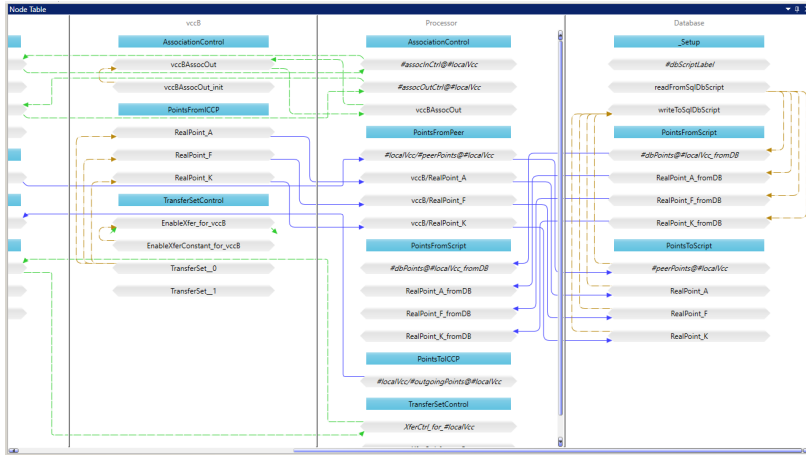
The design and organization of LEC is very much influenced by MMS. Much of the high-level organization finds its origins in MMS concepts. This means that many of the rules of LEC find their origin in the MMS specification. For example, MMS variable names can only consist of 32 characters and can only contain the alphanumeric characters and underscores (character '_'). This rule is enforced in LEC for explicit input and output ICCP points. For more information about MMS refer to the ISO's website.

VMD

LEC uses the MMS concept of Virtual Manufacturing Devices (VMD's) to represent component devices and systems. For example, a VMD on in an LEC Server configuration might represent a VCC (a VMD in the ICCP protocol) or a device from an entirely unrelated protocol, for example: A DNP master station, a Modbus slave, or an OPC UA server.

Each type of VMD in an LEC configuration is associated with a specific type of communication protocol or application interface, for example: ICCP, DNP, Modbus, ODBC, etc.

A VMD within an LEC Server configuration can connect to a device that exists outside of the LEC Server or to the other VMD's within the LEC server. VMD's allow LEC to capture, transform, and route data to devices, systems, or applications in a form that the other devices, systems, or applications can understand.



In the LEC Configuration Manager, VMD's are displayed as vertical rectangular panels. In the above figure, the LEC Configuration Manager is displaying three VMD's: vccB, Processor, and Database.

Configuration Aliases

The LEC Configuration Manager uses Configuration Aliases to organize and manage multiple LEC server configurations on the same machine. Each configuration created, edited, or viewed in LEC Configuration Manager is associated with a particular Configuration Alias. Typically, customers will only use one LEC Server instance per machine in production. By convention, that configuration alias is named **cfg**.

Configuration Manager VMD

Each LEC configuration contains at least one VMD called the Configuration Manager VMD. It represents the local machine (the machine running LEC) as a VMD within the configuration. This VMD is used to define certain nodes that represent variables in the LEC configuration that are related to the local machine, like a system variable or the status of a TCP connection. Configuration Manager VMD's are automatically created and named after the Configuration Alias and suffixed with the string **_LDSMGR**.

Points

Points are representations of measurements, states, signals, commands, or messages.

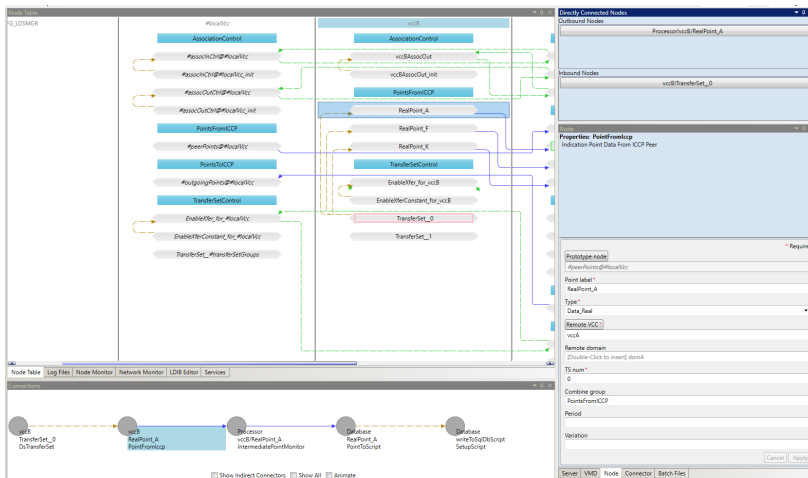
For example, your LEC Server configuration might have a point that represents the measured output of a power plant. Other points in your configuration might represent the temperature inside a transformer or the control signal to open a circuit breaker. Points live inside VMD's. LEC orchestrates where these points come from, where they go, and it applies any required logic, filtering, or transformations along the way.

Nodes

A node is the representation of a point at a particular step in its dataflow through the LEC Server. In an LEC configuration, a single point is often associated with a number of nodes.

For example, if there is a point on a SCADA Server that represents the current through a line, then that point might start in the LEC Server configuration as a PointFromDnp node, get filtered in the Processor VMD as a DeadbandFilter node, and finally leave the server as a PointToIccp node.

Every node in LEC exists as an in-memory variable. Nodes are displayed in the Connections panel, the Node Table, and the Node Monitor.



In the above figure, nodes are displayed as gray horizontal bars inside VMD's in the LEC Configuration Manager. The blue horizontal bars are just labels used to visually group nodes within a VMD called Combine Groups.

Connectors

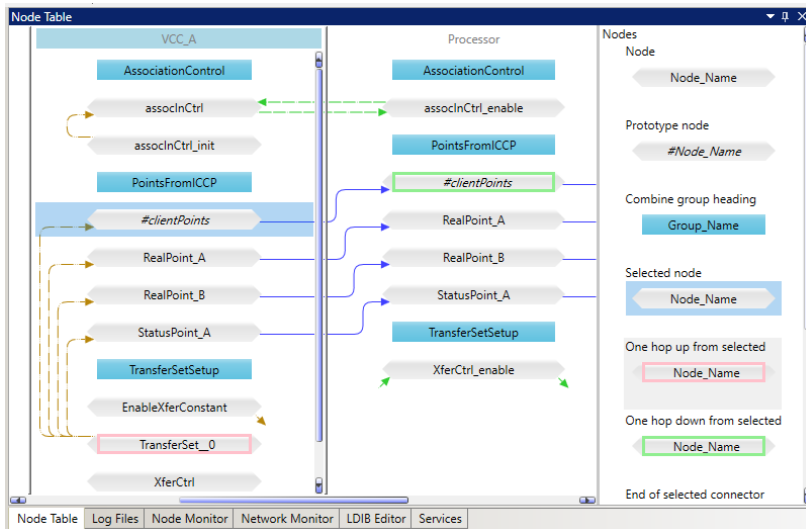
A connector connects two nodes in an LEC Server configuration. It defines under which circumstances and in which direction data should flow between two nodes. Connectors are displayed as arrows in the LEC Configuration Manger. The types of connectors are: Implicit, Update, Demand, and Two Way.

Filter Nodes

A filter node is a special type of node in the LEC server. It is used to transform data or apply logic to the flow of data through that node. An example of a filter is the DeadbandFilter node. It is one of the many built-in filters that ship with LEC. This filter propagates data past itself (further along in the dataflow) if and only if the value of the data has changed by some configurable amount or percentage.

Prototype Configurations

Most LEC server configurations perform the same operations on large numbers of similar points. It is not practical to individually specify each node in the LEC Configuration Manager. Instead, you can use a Prototype Configuration as a model of the actual configuration. In a prototype configurations, VMD's, nodes, and connectors in the configuration act as placeholders and templates. Prototype VMD's and prototype nodes have names that begin with a '#' character. At startup, the LEC Server populates all prototype



VMD's, nodes, and connectors with actual VMD's, nodes, and connectors using information provided to it in a batch file. The above figure shows the VCC_A VMD

that contains a prototype node named #clientPoints. The RealPoint_A node in VCC_A is generated by this prototype node from information included in a batch file.

Batch File

A batch file provides the information necessary to populate any prototype VMD's or prototype nodes with meaningful server configuration information.

Batch files are text files in CSV or JSON format that contain one or more tables. Each table has a header row with column values that correspond to various parameters of prototype VMD's and prototype nodes in an LEC prototype configuration. Every other row in the table specifies the exact value of these parameters for an actual VMD or node in the dataflow. Connectors are not explicitly specified within a batch file. They are automatically built based on the connections between prototype nodes in the prototype configuration.

PDI Macros

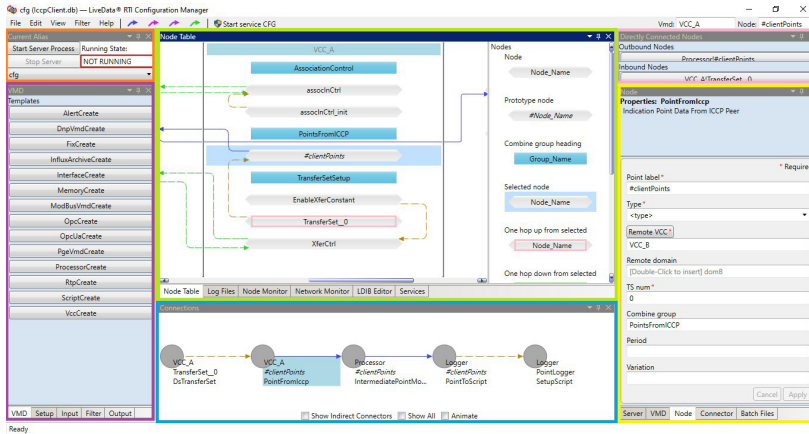
PDI macros, sometimes referred to as just macros in LEC Configuration Manager menus, are pieces of reusable Programmable Data Interface (PDI) code used to define an LEC configuration. PDI is a declarative language created to define how data should flow through the LEC server. The LEC Configuration Manager uses a collection of PDI macros defined in a file called DataflowMacros.pdi at server start-up to implement server configurations defined in the Configuration Manager. Before the LEC had a GUI-based configuration tool, text-based PDI files were used exclusively to define configurations.

Each type of node, connector, and VMD in the LEC Configuration Manager corresponds to a PDI macro defined in the DataflowMacros.pdi file or a macro defined in a customer-specific PDI file or Python file. The parameters shown in the LEC Configuration Manager Properties panel are used as inputs for these PDI macros.

LEC User Interface

The LEC Configuration Manager is a Windows desktop application that allows users to design, view, and edit LEC Server configurations, and monitor and manage a running LEC Server instance. The user interface includes a [menu bar](#) and six panels. Three of these panels have additional tabs used to access various tools and functions of the Configuration Manager.

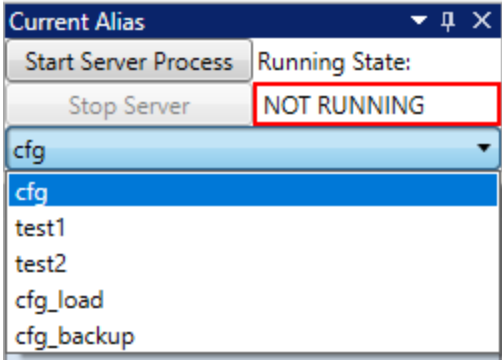
The following screenshot shows the six panels in the default view of the LEC Configuration Manager. In this image, the **VMD** tab is open in the **Templates** panel, the **Node Table** tab is open in the **Central** panel, and the **Node** tab is open in the **Properties** panel.



Current Alias Panel

The Current Alias panel displays the configuration alias currently selected, and allows you to switch between different aliases. Multiple configuration aliases can be created, though typically only one is used in a production environment. It also shows the running state of the Oracle Utilities Live Energy Connect (LEC) Server associated with the current configuration alias and lets you start and stop the server as a Windows process instead of running it as Windows service, which is useful in testing and troubleshooting scenarios.

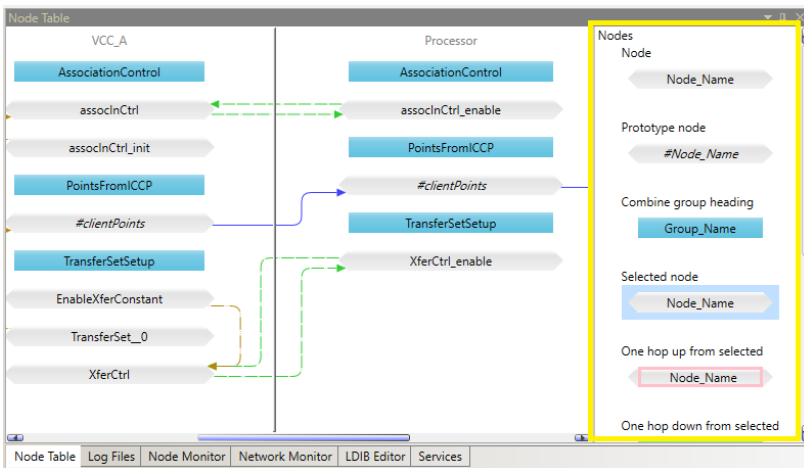
The following screenshot shows the **cfg** configuration alias currently selected. Click the name of a configuration alias in the drop-down menu to switch to a different alias. The **Running State** box shows the LEC server instance associated with this configuration alias is not running.



Central Panel

This panel has tools used to explore an Oracle Utilities Live Energy Connect (LEC) Server configuration and to monitor it if it is running.

The following image shows the **Node Table** tab open in a configuration with two VMD's. The **Node Table** will display VMD's, the nodes inside them, connectors between the nodes, and combine group labels. The legend at the right describes how items are displayed in the Node Table.



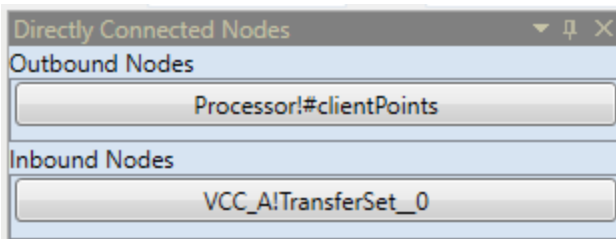
You can select tabs at the bottom of the central panel to change what is displayed. Each tab provides the information about LEC configuration:

- **Node Table** - Contains a visual representation of the LEC Server configuration. Use it to view, build, and edit the configuration. Select VMD's, nodes, and connectors from the Node Table to edit their properties in the Properties panel.
- **Log Files** – Displays the running log of the LEC server.

- **Node Monitor** - Allows to monitor the current values of certain nodes in real-time. This is useful for testing and troubleshooting.
- **Network Monitor** - Shows the status of MMS and ICCP associations for each VMD in the configuration.
- **LDIB Editor** - Displays the MMS and ICCP association parameters table for VMD's.
- **Services** - Lists the LEC configuration aliases that are registered as Windows services and the status of each service.

Directly Connected Nodes Panel

This panel displays the full name of the input and output nodes attached to the currently selected node by connectors. You can use it to select a node connected to the currently selected node if it is not easy to select them from the **Node Table** in the **Central** panel or from the map in the **Connections** panel.



Properties Panel

This panel specifies the properties of selected VMD's, nodes, and connectors in the Oracle Utilities Live Energy Connect (LEC) Server configuration. It also specifies batch files to be loaded and some parameters that need to be specified for LEC Server to start.

There are various VMD, node, and connector types. Each type has its own set of properties. To identify what a field in the Properties panel is used for, hover over the field to read its tool tip. Fields that are required to create a particular VMD, node, or connector are marked with a red asterisk.

The following image shows the Properties panel when a PointFromIccp node is selected. Most nodes have Point label and Type parameters.

Selecting an item in the configuration (by clicking a node or VMD in the Node Table) automatically opens the relevant Properties panel for that item.

In the **Properties** panel, you can switch between the following tabs:

- **VMD** - Specify the properties of a selected VMD in the configuration. Each type of VMD has its own defined set of parameters.
- **Node** - Specify the properties of a selected node in the configuration. Some commonly used node types are: IntermediatePointMonitor, PointToIccp, PointToDnp, and PointFromMemory. Each type of node has its own defined set of parameters.
- **Connector** - Specify the properties of a selected connector in the configuration. The types of connectors are: UpdateConnector, TimedConnector,

DemandConnector, and TwoWayConnector. Each type of connector has its own defined set of parameters.

- **Batch Files** - Allows to load and unload batch files into an LEC server configuration. It also shows which batch files are currently loaded in the open configuration.
- **Server** - Specify server-specific information for the Live Energy Connect Server

Note: By default, the LEC server listens on LOCALHOST (i.e. 127.0.0.1) for incoming MMS/ICCP associations. If you're configuration requires inbound MMS/ICCP associations from remote peers (i.e. VCC's on other machines), you will need to modify your LEC configuration to allow the server to listen on the appropriate external interface. To do this, with the appropriate configuration alias selected, open the **Server** tab of the **Properties** panel. In the **Extra params** field, add `"/listen=<ip_address>"` where `<ip_address>` is the IP address of the network interface you want the server to listen on (e.g. `"/listen=192.168.1.10"`).

In the following image, the value in the **Extra params** field tells the LEC Server to listen for inbound ICCP associations from remote ICCP peers on the interface with the IP address of 192.168.1.10."

Server [Window Title]

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 *

Soap port *

Mms listen ip

Mms port *

Global flags

User macros

Extra params

Timestamp

Alter Registry

Cancel Apply

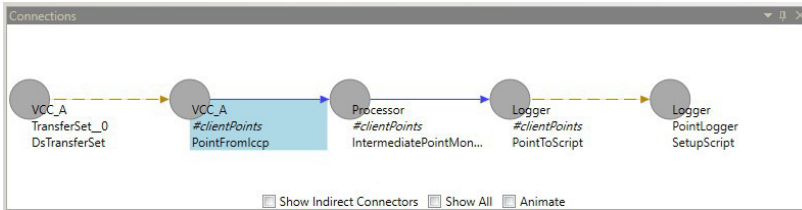
Server VMD Node Connector Batch Files

Connections Panel

This panel displays a map of the selected node and the nodes connected to the selected node. It is used to visually represent the dataflow or path of a given point through the Oracle Utilities Live Energy Connect (LEC) Server. Each node in the map will be listed with the name of its VMD, its own name, and the type of node it is.

You can select any node that appears in the Connections panel. It can be useful to select nodes in a configuration from the Connections panel so that you don't have to search for them in the Node Table.

In the Connections panel, you can specify a limit on how close connected nodes have to be to the selected node in order to appear in the map.



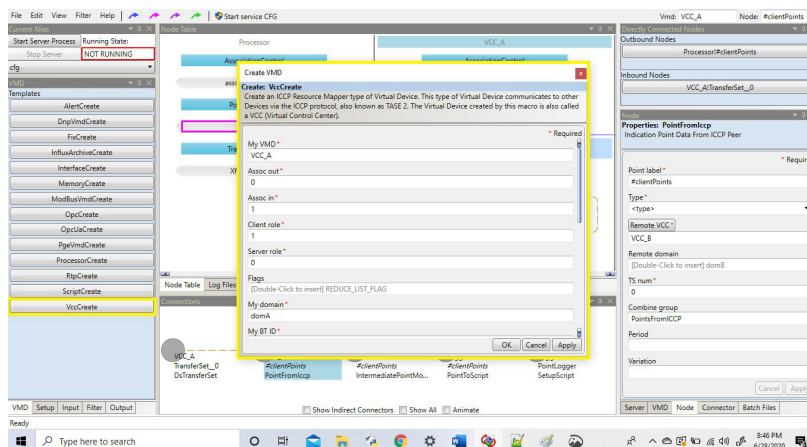
The above screenshot from LEC Configuration Manager shows the map of the dataflow around a selected node and its connected nodes in the Connections panel. The #clientPoints inside VCC_A VMD is the selected node.

Templates Panel

This panel is used to add new VMD's and nodes to your Oracle Utilities Live Energy Connect (LEC) Server configuration. It contains a library of VMD and node templates. The types of node templates available are organized into four tabs: **Setup**, **Input**, **Filter**, and **Output**.

You can browse through the tabs in this panel to find the type of VMD or node you are looking to create. Select an item to launch a Create window that will allow you to specify the appropriate Properties and create the object.

In the following image, the VccCreate option was selected from the VMD tab in the Templates panel so the Create VMD tool for a VccCreate VMD was launched. By default, each field will be populated with the last value specified for that field.

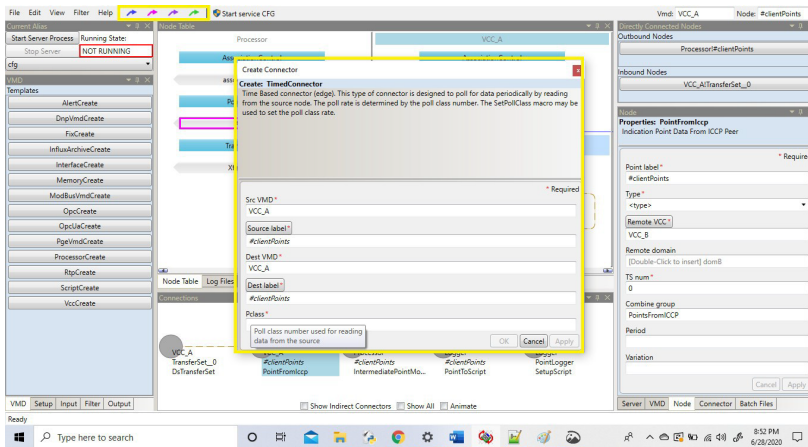


From the Templates panel, you can create items from the following tabs:

- **VMD** - Lists the types of VMD's that can be created in the configuration. Some common VMD types are: ProcessorCreate, VccCreate, and ScriptCreate.
- **Setup** - Lists the setup nodes types available for a selected VMD. The nodes listed in this tab will differ depending on what type of VMD is currently selected before clicking the tab. Setup nodes are often nodes that give the LEC server access to variables that control associations (or connections) between devices or the organization of points inside a device
- **Input** - Lists the input node types available for a selected VMD. The nodes listed in this tab also differ depending on what VMD is selected before selecting the tab. Input nodes are used to specify data coming into the VMD.
- **Filter** - Lists the types of filter nodes available for a selected VMD. Filter nodes allow you to apply transformations to the points in your dataflow or apply rules to their flow through the configuration. LEC comes with built-in filter nodes that provide a number of different functions, such as deadbanding, scaling, and quality filtering.
- **Output** - Lists the types of output nodes available for a selected VMD. The nodes listed in this tab also differ depending on what VMD is selected before selecting the tab. Output nodes are used to specify data leaving the VMD.

Menu Bar

The menu bar organizes the remaining tools in the LEC Configuration Manager and provides quick access to some of the most used features of the software. It also contains a group of buttons used to create connectors between nodes and an easy-to-find button that can be used to stop or start the LEC server.



Again, in addition to providing access to the **File**, **Edit**, **View**, **Filter**, and **Help** menus, the **Menu Bar** consists of buttons that launch the Create Connector tool and allow you to create new connectors. The figure above displays the Create Connector window for creating a TimedConnector.

The following table gives you an overview of each menu bar item.

Item	Options
File	<ul style="list-style-type: none"> ■ Open, import or save server configurations ■ Create new configuration aliases ■ Create new Windows services for a configuration alias ■ Load or unload batch files ■ Reload macros
Edit	<ul style="list-style-type: none"> ■ Delete selected VMD's, nodes, or connectors ■ Copy and paste selected VMD's
View	<ul style="list-style-type: none"> ■ Hide and unhide certain tools in the Configuration Manager ■ Restore the default layout of the Configuration Manager ■ Open the Variable Access Tool
Filter	<ul style="list-style-type: none"> ■ Specify which VMD's and nodes are visible in the Node Table tab, Node Monitor tab, and Connections panel

Item	Options
	<ul style="list-style-type: none"> ▪ Hide and unhide VMD's
Help	<ul style="list-style-type: none"> ▪ Find the which version of LEC is installed ▪ Gather diagnostic data for troubleshooting and support
Connector	<p>Use the Connector buttons to create connectors. A connector is represented by an arrow that starts at a source node and ends at a destination node. The types of connectors available are:</p> <ul style="list-style-type: none"> ▪ An UpdateConnector subscribes to receive notifications from the source node. In other words, when the value of the source node changes, the destination node is updated. UpdateConnectors are displayed as blue arrows. ▪ A TimedConnector polls for data from the source and updates the destination periodically. Timed connectors are associated with an LEC server poll class that defines the polling period. Poll classes are defined as setup node in the configuration VMD. TimedConnectors are displayed as pink arrows. ▪ A DemandConnector reads from the source node whenever another connector or device reads from the destination node. DemandConnectors are displayed as purple arrows. ▪ A TwoWayConnector subscribes to receive notifications from the source node and the destination node. In other words, when the value of the either node changes, the other node is updated. TwoWayConnectors are displayed as green arrows. ▪ An ImplicitConnector is a connector created automatically by the Configuration Manager when using one node as a parameter input to another node.

Item	Options
	ImplicitConnectors are displayed as brown arrows.
Start / Stop Service	Use the Start service or Stop service button to start or stop the LEC service for the currently selected configuration alias.

A Quick Configuration Manager Tutorial

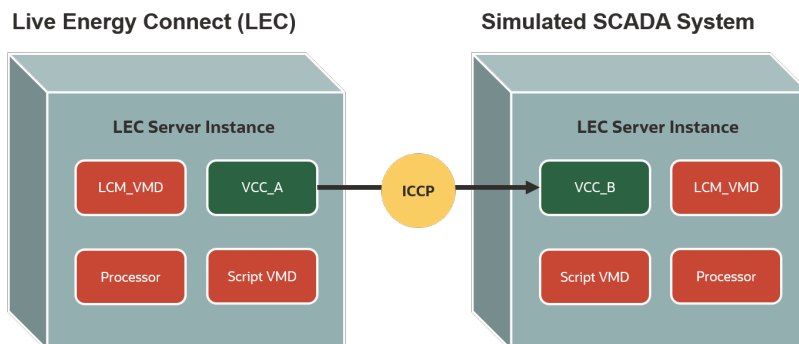
This tutorial uses two provided example configurations to get familiar with using the Oracle Utilities Live Energy Connect (LEC) Configuration Manager as quickly as possible. For a more detailed tutorial, [see Creating an ICCP Client](#).

Scenario for the Example ICCP Client Configuration

The goal of this tutorial is to use Oracle Utilities Live Energy Connect (LEC) to get information over ICCP from a remote SCADA system. For more information on the ICCP protocol, see [ICCP Reference](#).

The main LEC Server configuration used in this tutorial acts as an ICCP client. The simulated SCADA system acts as an ICCP server. In this configuration, the LEC Server receives information about points on the remote SCADA system and logs the information to a file. For details about creating the ICCP client configuration, see [Creating an ICCP Client](#).

The following diagram shows the systems involved in this tutorial. In this scenario the main LEC Server configuration (left) acts as an ICCP client. It makes an outbound association to the remote SCADA system that acts as an ICCP server.



To simulate the remote SCADA system, we will use another LEC server instance on the same machine. This simulated SCADA system will generate random data values using a Script VMD that hosts a Python script called **Volts.py**.

Note: These configurations are provided in the example files that come with the LEC installation. In addition to this tutorial, there are other example configuration and batch files (with READ ME's). These examples are available in the **C:\ProgramData\LiveEnergyConnect** directory. Some of these examples demonstrate using LEC to communicate with devices over other protocols like DNP or Modbus, or as an interface to other systems like a SQL Server database.

Most customers do not need to create their own LEC Server configurations. Instead, they are provided with configurations by Oracle Utilities and will only use the Configuration Manager to edit them or monitor the customer's LEC Server instances.

Importing Oracle Utilities Live Energy Connect Server Configurations

Oracle Utilities Live Energy Connect (LEC) Server configurations are saved as SQLite files with **.db** extensions. Most configuration files define a prototype configuration that, when loaded with one or more batch files, will define a complete and useable server configuration.

Importing an LEC Server Configuration

1. Open LEC Configuration Manager as an Administrator.
2. Select **Import configuration** from the **File** menu.
3. Navigate to the **C:\ProgramData\LiveEnergyConnect** directory in the file browser that opens.
4. Select the **lccpClient.db** file and select **Open**.

The provided ICCP client example configuration has four VMD's:

- CFG_LD SMGR (the Configuration Manager VMD)
- VCC_A (a VCC acting as an ICCP client)
- Processor (a Processor VMD)
- Logger (a Script VMD)

Loading a Batch File

The Oracle Utilities Live Energy Connect (LEC) Configuration Manager loads one or more batch files into a prototype configuration. After the batch files are loaded, the Configuration Manager has the information needed to define a specific, useable server configuration. Batch files are text files in CSV or JSON format that specify lists of point names and parameters. Sometimes batch files also contain lists of VMD names and parameters if the Prototype Configuration uses prototype VMD's.

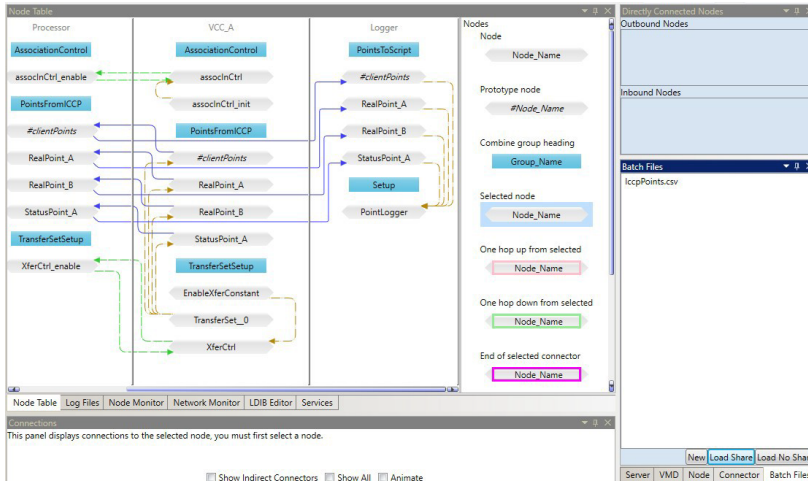
Typically, Oracle Utilities engineers will help you create your prototype configuration and batch files.

To load a batch file:

1. Select **Load batch (share)** from the **File** menu.
2. Navigate to the **C:\ProgramData\LiveEnergyConnect\Config** directory in the file browser that opens.
3. Choose the **IccpPoints.csv** batch file and select **Open**.
4. The file will load and appear in the **Batch Files** tab in the **Properties** panel (lower right).

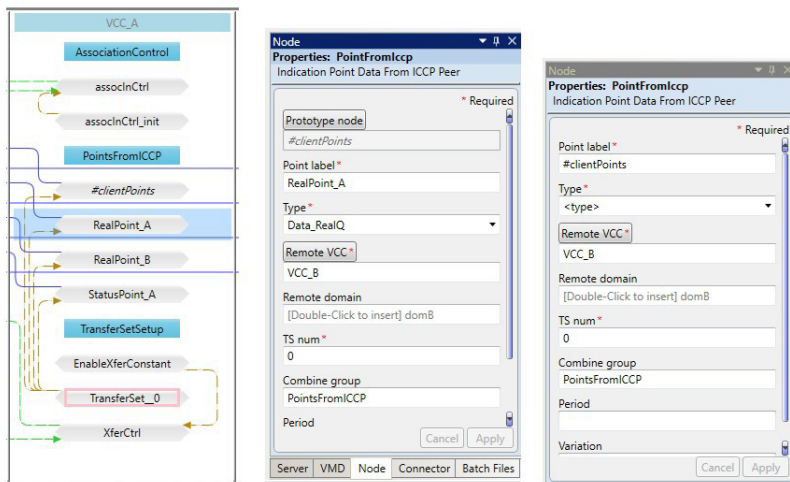
Note: Most server configurations use batch files that are loaded using **Load batch (share)**. The other option, **Load batch (no Share)** is used less often. **Load batch (share)** treats a collection of batch files as a single batch file (the tables in one batch file can reference something defined in an earlier loaded batch file). If loading only one batch file, you can use either option.

The following image shows what the example configuration looks like in the Configuration Manager after loading the **IccpPoints.csv** batch file. The prototype nodes (the nodes whose labels start with the '#' character) have generated new nodes based on the information in the batch file. In this example, only nodes were created, but batch files can also be used to build VMD's from prototype VMD's as well.



After loading the IccpPoints.csv batch file, the Configuration Manager generates new nodes based on the prototype nodes in the configuration. For example, in VCC_A, the node RealPoint_A is generated from the prototype node #clientPoints.

Consider the nodes in the VCC_A VMD that were generated by the batch file. VCC_A includes a prototype node #clientPoints. The figure below shows the properties of this node. Click the node in the Node Table and select the Node tab in the Properties panel to view the nodes properties.

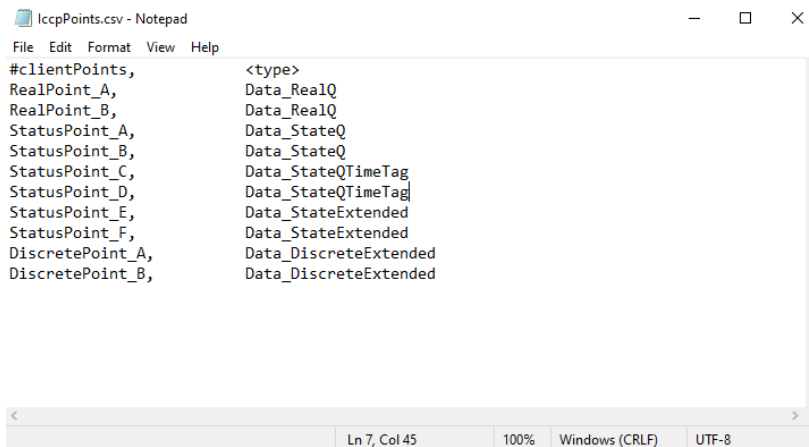


The Configuration Manager generates the node RealPoint_A from the placeholder parameters for the

Point label and **Type** parameters specified in the prototype node #clientPoints."

The figure below shows the structure of the IccpPoints.csv batch file. Point label and Type are common parameters for many types of nodes. In a batch file, each row after

the header row corresponds to a node (or VMD) that should be generated. Each column corresponds to a parameter of prototype node (or prototype VMD) in the prototype configuration.



```
IccpPoints.csv - Notepad
File Edit Format View Help
#clientPoints, <type>
RealPoint_A, Data_RealQ
RealPoint_B, Data_RealQ
StatusPoint_A, Data_StateQ
StatusPoint_B, Data_StateQ
StatusPoint_C, Data_StateQTimeTag
StatusPoint_D, Data_StateQTimeTag
StatusPoint_E, Data_StateExtended
StatusPoint_F, Data_StateExtended
DiscretePoint_A, Data_DiscreteExtended
DiscretePoint_B, Data_DiscreteExtended
Ln 7, Col 45 100% Windows (CRLF) UTF-8
```

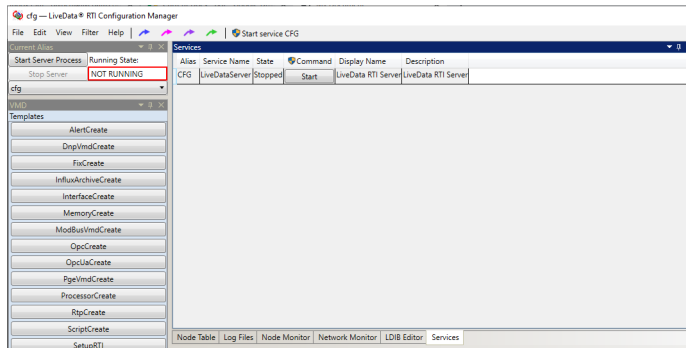
In the example configuration, the Type parameter refers to an ICCP type like Data_ RealQ or Data_StateTimeTag but in other server configurations, valid values for Type might be an MMS type like <integer:32>, a type native to another industry protocol, or even a custom type defined by the user's configuration. For more information about ICCP types, refer to the [ICCP Reference](#).

Starting and Stopping the Live Energy Connect Server

After loading the batch files necessary to create all the nodes in the example configuration, you can start the Oracle Utilities Live Energy Connect (LEC) Server. Usually, the server is run as a Windows service. By default, the cfg configuration alias will already have a Windows service registered after installing LEC.

To start the LEC Server as a Windows service:

1. Click the **Start Service CFG** command in the menu Bar. Alternatively, click **Start** from the **Service** tab of the **Central** panel.



2. If you are not running the Configuration Manager with Administrative privileges, a pop-up message will display asking if you want to allow the **servxnt.exe** program (the server executable) to make changes to the computer. Select **Yes**.

Now we can stop the server and work on setting up the SCADA system simulator configuration:

- To stop the server, click the **Stop Service** command on the **Menu Bar** at the top of the Configuration Manager window. Alternatively, click **Stop** in the **Services** tab of the Configuration Manager.

Creating a New Configuration Alias

You will need to set up the server configuration that will simulate a remote SCADA system. To do this we will create a new configuration alias in Oracle Utilities Live Energy Connect (LEC) Configuration Manager to work with new configuration:

1. Select **New alias** from the **File** menu.
2. In the **Create new alias** tool, type **ScadaSim** for the new configuration alias.
3. Select **OK**. The Configuration Manager will switch to the new configuration alias and show an empty configuration.

Loading the Simulated SCADA System Configuration

1. Select **Import configuration** from the **File** menu.
2. Navigate to the **C:\ProgramData\LiveEnergyConnect** directory in the file browser.
3. Select **lccpServerSim.db** from the list and click **Open**.

Loading a Batch File for a Simulated SCADA Server Configuration

1. Select **Load batch (share)** from the **File** menu.
2. Navigate to the **C:\ProgramData\LiveEnergyConnect\Config** directory in the file browser that opens.
3. Select **IccpSimPoints.csv** from the list and click **Open**.

Adjusting the Server Parameters for the Simulated SCADA Server Configuration

The LEC server needs to have two ports specified at startup to run successfully: a port for communication between the Configuration Manager and the server (using the SOAP protocol) and a port for communication between VMD's inside and outside of the server (using MMS). The default SOAP port is 8089 and the default MMS port is 102.

Most customers will not need to adjust these ports because they will only run one of Oracle Utilities LiveEnergy Connect instance on a single machine. However, in this example tutorial, we will need to change the ports used by the instance simulating the SCADA server configuration because the two instances can't use the same port on the same machine at the same time.

The screenshot shows a 'Server' properties dialog box. The title bar says 'Server'. Below the title bar, it says 'Properties: LiveData RTI Server startup parameters. (Note if you alter the registry, then when the altered service runs, it will run with this configuration)'. The dialog contains several input fields: 'App name*' with the value 'ScadaSim'; 'Soap port*' with the value '8090'; 'Mms listen ip' with the value '0.0.0.0'; 'Mms port*' with the value '103'; 'Global flags' with the value '1'; 'User macros' (empty); 'Extra params' (empty); and 'Timestamp' with the value '4/9/2020 5:09:10 PM'. There is a checkbox labeled 'Alter Registry' which is currently unchecked. At the bottom right of the dialog are 'Cancel' and 'Apply' buttons. Below the dialog is a tabbed interface with tabs for 'Server', 'VMD', 'Node', 'Connector', and 'Batch Files'. The 'Server' tab is currently selected.

To adjust the server parameters for the SCADA server simulator configuration:

1. Select the **Server** tab in the **Properties** panel.
2. Change the Soap port from **8089** to **8090**.
3. Change the Mms port from **102** to **103**.
4. Click **Apply**.

Creating a Service for the Simulated SCADA Server Configuration

The Windows service for the simulated SCADA server configuration has not been created yet. The Services tab in the Central panel only shows one service cfg listed for the configuration alias.

To create a Windows service for the SCADA server simulator configuration:

1. Select **Create Service for Alias** from the **File** menu.
2. Type a name in the **Service Name** field. This name is used to register the service.

Note: Windows service names cannot include spaces.

3. In the **Display Name** field, enter a display name for the service, which can contain spaces.
4. Select the startup behavior the service should have in the **Startup Type** field. Let's select **Manual** since this is just an example tutorial.
5. Add a description of the service in the **Description** field. This description will appear in the **Services** tab of the Configuration Manager and the Windows Services Manager tool.
6. Click **OK**.

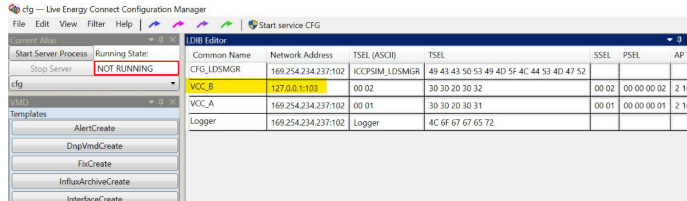
Adjusting the Network Information Using the LDIB Editor

Now we will use the LDIB Editor in the Central panel to specify the address of the remote VCC for each server configuration. In this case for both server configurations, the remote ICCP peer's IP address will be the LOCALHOST address, 127.0.0.1, because we are running both configurations on the same machine.

Note: If we were using a prototype VMD to create VCC_A and VCC_B then we could specify this information in a batch file.

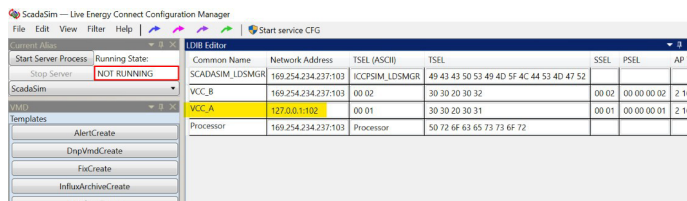
To update the Cfg configuration's remote peer's IP address in the LDIB Editor:

1. Select the **cfg** alias from the **Configuration Alias** drop-down menu.
2. Open the **LDIB Editor** tab in the **Central** panel.
3. For the row associated with VCC_B in the **Common Name** column, change the value for **Network Address** to **127.0.0.1:103**.



To update the ScadaSim's configuration's remote peer's IP address in the LDIB Editor:

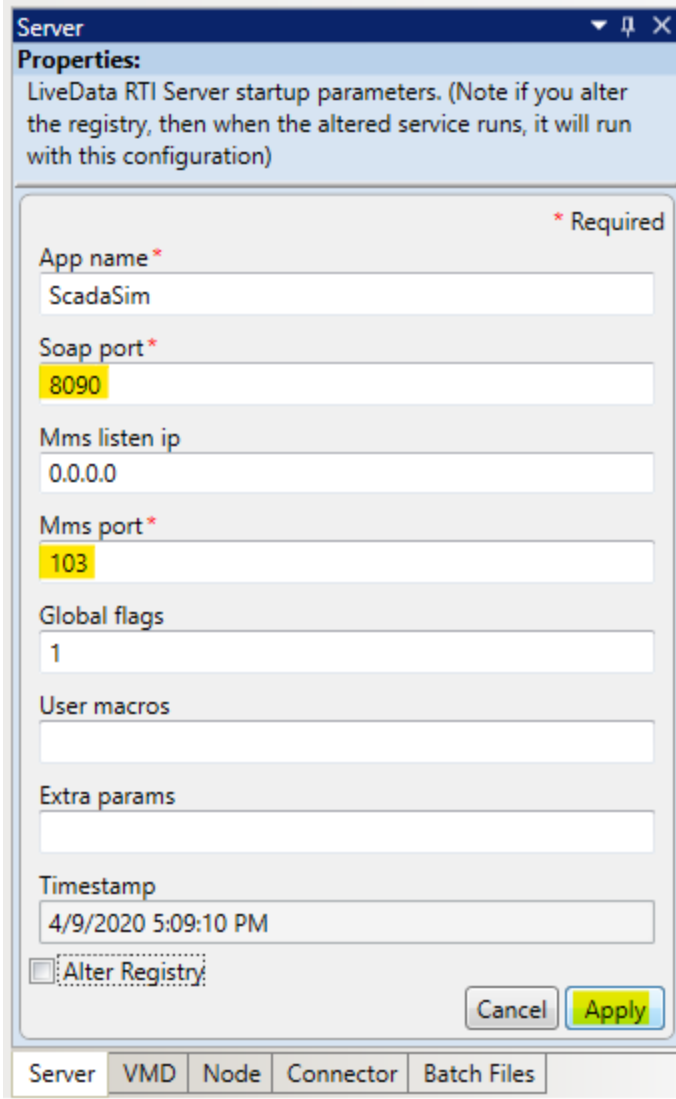
1. Select the **ScadaSim** alias from the **Configuration Alias** drop-down menu.
2. Open the **LDIB Editor** tab in the **Central** panel.
3. For the row associated with VCC_A in the **Common Name** column, change the value for **Network Address** to **127.0.0.1:102**.



Adjusting the Server Parameters for the Simulated SCADA Server Configuration

The LEC server needs to have two ports specified at startup to run successfully: a port for communication between the Configuration Manager and the server (using the SOAP protocol) and a port for communication between VMD's inside and outside of the server (using MMS). The default SOAP port is 8089 and the default MMS port is 102.

Most customers will not need to adjust these ports because they will only run one of Oracle Utilities LiveEnergy Connect instance on a single machine. However, in this example tutorial, we will need to change the ports used by the instance simulating the SCADA server configuration because the two instances can't use the same port on the same machine at the same time.



To adjust the server parameters for the SCADA server simulator configuration:

1. Select the **Server** tab in the **Properties** panel.
2. Change the Soap port from **8089** to **8090**.
3. Change the Mms port from **102** to **103**.
4. Click **Apply**.

Additional Tools and Features in LEC Configuration Manager

This section describes additional tools and features in the Oracle Utilities Live Energy Connect (LEC) Configuration Manager used to monitor and edit server configuration.

Live Energy Connect Logging

The Oracle Utilities Live Energy Connect (LEC) Server writes to a log file called **LIVEDATA.LOG**. You can view the log and configure the information included in the log in the **Log File** tab in the **Central** panel of the Configuration Manager.

Viewing Log Files

View the current log file from the **Log File** tab in the **Central** panel. Alternatively, use a text editor to view the log. To view in any text editor, navigate to **C:\ProgramData\LiveEnergyConnect\Logs** and open the **LIVEDATA.LOG** file. Also, if you double-click the bar in the **Log File** tab that contains the file path, Windows will open the current log in your default text editor.

Specifying the Logging Levels

Log messages can be divided into two major categories: Manager and Transport. Each category has five logging levels, which can be adjusted while the server is running to customize what information is written to the log.

Transport log messages contain information about the network connections the LEC Server configuration uses (TCP sockets, network addresses, routing, etc.). Unless you are troubleshooting a network connectivity issue, the **Transport Level** log messages are not useful.

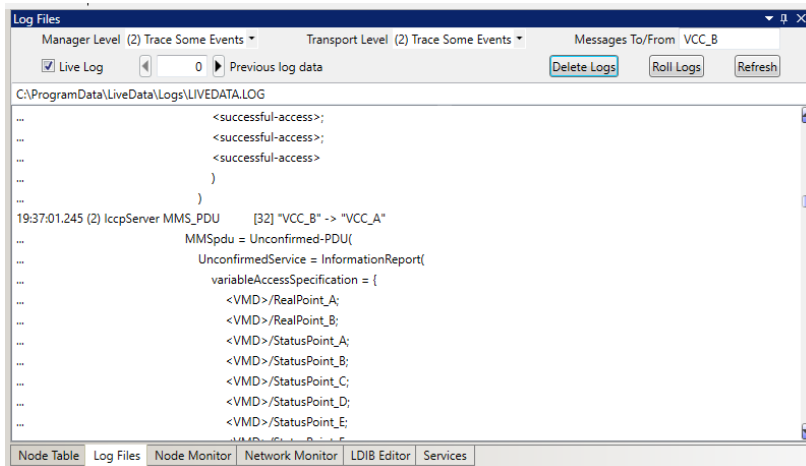
Manager log messages contain information about the LEC Server and its configuration (ICCP associations, protocol messages, etc.). The various sources of information for **Manager** log messages are divided into logging modules. Logging modules represent functional components of the LEC Server. Adjust which logging modules are associated with which logging level by editing the server's **svrxnt.ini** file. The default logging level used by each logging module is printed to the log file when the application starts.

The five logging levels are:

- (0) Errors
- (1) Diagnostics
- (2) TraceSomeEvents
- (3) TraceAllEvents
- (4) TraceAllData

Use the **Messages To/From** field to specify to view detailed MMS/ICCP information about a particular VMD. For example: If the Manager level is set to **2** and a remote VMD is specified in the **Messages To/From field**, then you can view the decoded MMS packets sent by the remote VMD.

The following image shows the log file produced when running the example LEC Server configuration used in the previous tutorial. There are decoded MMS packets (MMS PDU's) that appear in the log because the **Manager Level** is set to **2** and the remote VCC called **VCC_B** is specified in the **Messages To/From** field.



Rolling the Log Files

The higher the log level, the more verbose the logging. For example, a log level of **0** will only send error messages to the log, but a log level of **4** will send all available information to the log.

The more verbose a log file is, the faster it will grow. Log files automatically roll over based on the **NumOldLogs** and **MaxLogFileSize** parameters specified in the LEC Server file, **srvxnt.ini**. Select **Roll Logs** to roll over the logs manually from the Configuration Manager.

Managing the Log Files

Unselect **Live Log** to freeze the viewing panel of the log file in the **Log File** tab of the Configuration Manager. This is helpful if the log file is logging a lot of information but you want to review a small section of the log.

Select **Delete Logs** to delete all automatically rolled-over log files and the current log file. If the live logging feature is disabled or if the log files have been deleted, refresh the Configuration Manager's view of the current **LIVEDATA.LOG**. Select **Refresh** in the **Log File** tab to refresh the file.

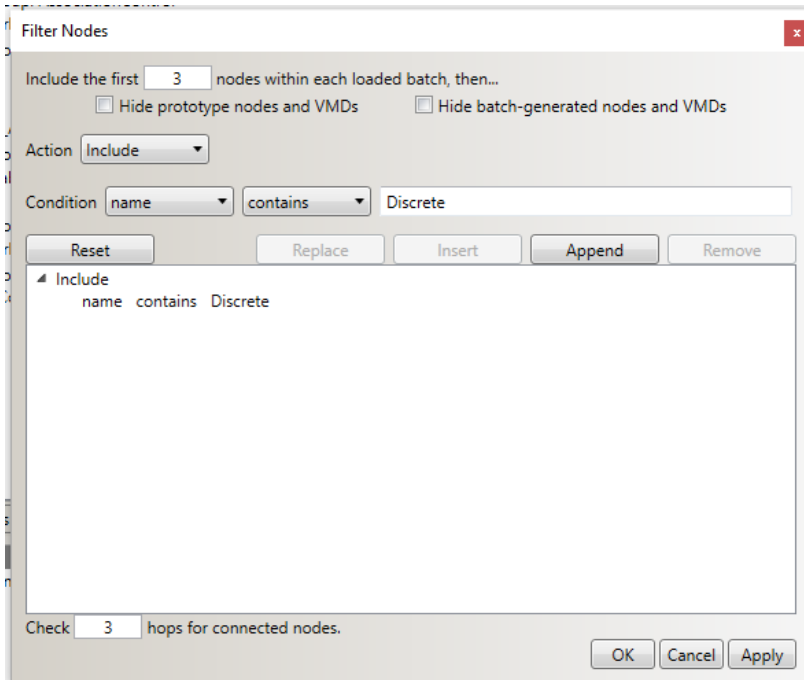
Specifying the Visible Nodes in the Configuration Manager

Let's say you want to monitor a specific node in the Oracle Utilities Live Energy Connect (LEC) Server configuration. By default, the node you are interested in may not show up in the **Node Table** or **Node Monitor** because the Configuration Manager limits the number of nodes displayed to keep the screen from getting cluttered. The Configuration Manager also limits the number of connections displayed in the **Connections** panel to keep the graph of connections that it displays readable. You can change what nodes and connections are visible with the Filter Nodes tool.

To create a new filter rule with the Filter Nodes tool:

1. Navigate to the **Filter** menu and select **Filter Nodes**.
2. Add a rule and change the drop-down options on the **Container** row to **name** and **contains**, respectively.
3. In the **Condition** text box type the whole node label of the node that you're interested in, or just a part of it and select **Append**. Now, nodes that contain that string in the node label is visible in the **Node Table** and **Node Monitor**.

The following image shows **Filter Nodes** tool containing a filter rule that makes nodes whose node labels contain the string **Discrete** visible in the **Node Table** and **Node Monitor**. Notice the number of hops (connections) from a selected node in the **Connections** panel is set to **3**, which is the default.



Note: VMD names and node labels are case-sensitive and so any strings specified in the Filter Nodes Tool are also case-sensitive.

To Adjust the connection hops displayed in the Connections panel:

1. Navigate to the **Filter** menu and adjust the value of the **Check hops for connected nodes** field.
2. Select **Append**. Now, the graphic in the Connections panel shows nodes within the desired number of connections from the selected node.

Variable Access Tool

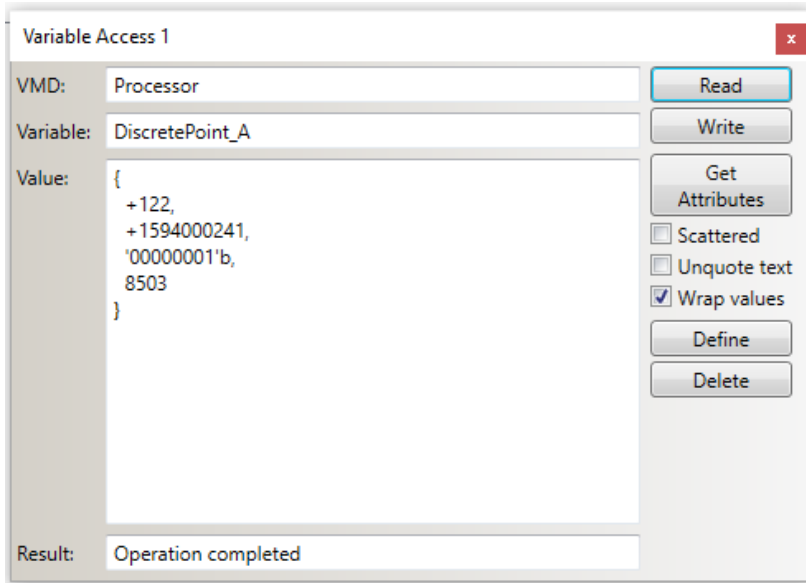
In addition to using the **Node Monitor** tab in the **Central panel**, you can monitor nodes in the Configuration Manager using the Variable Access tool. The Variable Access tool will also let you read and write values to certain points even if they are currently filtered out of the views in the **Node Monitor** and **Node Table** tabs.

Certain types of nodes are viewable and writeable. Typically, you will only be using the Variable Access tool to read or write IntermediatePointMonitor nodes.

To open the tool, select the **Variable Access** option from the **View** menu on the menu bar. To read a value from and write a value to a node with the Variable Access tool do follow these steps:

1. In the **VMD** field, select the name of the VMD that contains the node of interest like, for example, **Processor**.
2. Enter the name of the node to view in the **Variable** field. For example, if you are using the example configurations from the [Quick Configuration Manager Tutorial](#), start typing **DiscretePoint_A** and the **Variable** field will automatically populate the drop-down with possible nodes in the VMD that contain that string. Select the node you would like to read. For example, select **DiscretePoint_A** from the drop-down of the **Variable** field.
3. Select **Read**. If it exists, the current value for that point will display as a structure in LEC Server's MMS-DL syntax format. If it does not exist, an error message appears in the **Result** field.
4. To change the value of the node, edit the **Value** section of the Variable Access tool and select **Write**. If the write is successful, an *Operation completed* message appears in the **Result** field. If it not, an error message will appear in the **Result** field.

The following image shows the Variable Access tool after reading the IntermediatePointMonitor node in the processor VMD called **DiscretePoint_A** using the example ICCP client server configuration. This point is of type **Data_DiscreteExtended**, so its value includes four components: a discrete value (+122), a timestamp (+1594000241), a bit string of ICCP quality flags ('00000001'b), and a change-of-value counter (8503).



Note: Since the value of a node needs to be in a specific format, it is best to read the value of a node and use the result as a template to write to the node. This will avoid formatting errors.

Writing a value to a node with the Variable Access tool changes the value of that node in the server. This will affect the rest of the configuration. Writing values with the Variable Access tool is useful when developing or debugging server configurations, but it should be done with caution in production environments. In production environments, the side effects of a write to a particular node in an LEC Server instance should be well understood by the Configuration Manager user.

LDIB Editor

The **LDIB Editor** tab in the **Central panel** of the Configuration Manager contains networking information and some MMS/ICCP information that the Oracle Utilities Live Energy Connect (LEC) Server uses to establish MMS/ICCP associations between VMD's. The information in the LDIB Editor is automatically generated based on the information in the configuration and batch files and can be edited.

Note: Editing information in the LDIB Editor will not write that information to a batch file. So, if a VMD's networking information is specified in a batch file, any changes to that information made from

the LDIB Editor will not persist if the batch file is unloaded and reloaded into the configuration.

There are 13 fields included in the LDIB Editor table for each VMD (local and remote). Each VMD in your configuration may have a value for some or all of the fields listed on the following table:

LDIB Editor Field	Description
Common Name	Name for the VMD; only used by LEC
Network Address	IP address and TCP port of the MMS/ICCP interface (MMS port defaults to 102)
TSEL (ASCII)	2-byte or 4-byte number used to select the correct instance of the transport layer in ASCII format
TSEL	2-byte or 4-byte number used to select the correct instance of the Transport layer as hexadecimal number
PSEL	2-byte or 4-byte number used to select the correct instance of the Presentation layer
SSEL	2-byte or 4-byte number used to select the correct instance of the Session layer
AP Title	Optional object identifier representing the Application Process Title given to this application
AE Qualifier	A long integer (32-bit, signed int) is used to qualify the application entity
IS	Select/Deselect International Standard (Defaults to selected and is mutually exclusive with DIS field)
DIS	Select/Deselect Draft International Standard (Defaults to

LDIB Editor Field	Description
	unselected and is mutually exclusive with IS field)
Secure ICCP	Enable/Disable Secure ICCP (Defaults to disabled)
Monitoring	Enable/Disable monitoring of node value in the Configuration Manager(Defaults to enabled)
UCA	Enable/Disable Utility Communications Architecture (Defaults to disabled)

Viewing Network Connection and ICCP Association Configuration Information in the LDIB Editor

If you changed your configuration or loaded a batch file since last viewing the LDIB editor, click **Refresh** to view the current values. Use the LDIB Editor to check and verify your configuration's ICCP associations by looking at the **TSEL**, **SSEL**, **PSEL**, **AP Title**, and **AE Qualifier** columns.

Changing Network Connection Information

To change the IP address of a VMD or an ICCP Association parameter, it is recommended to change it in the **VMD** tab of **Properties** panel in the Configuration Manager or the batch file if one was used to define the VMD and not the LDIB Editor. However, it can be changed in the **Network Address** column of the LDIB Editor. After making any changes, click **Apply**. Any changes applied to the **LDIB Editor** table will override values specified elsewhere in the configuration or batch files. To restore these values if you can, re-specify the VMD parameters in the Properties panel for the given VMD or reload the batch files containing the relevant information.

Writing an LDIB.INI File

If the configuration needs to communicate with an external application using Microsoft's ActiveX (OCX) controls, the server configuration will need an LDIB.ini file that contains the remote application's VMD's association information.

Clicking **Write LDIB.ini** exports the information shown in the LDIB Editor table as the **LDIB.ini** file. However, most customer's LEC server configurations do not require an LDIB.ini file.

Gathering Diagnostic Information for Support

If there are issues in configuring or running Oracle Utilities Live Energy Connect (LEC), contact [My Oracle Support](#) (MOS) by creating a service request from the support portal. You can gather information that will be helpful to MOS engineers using the Configuration Manager's **Gather Diagnostic Data** tool.

To gather diagnostic data from the Configuration Manager:

1. Select **Gather Diagnostic Data** from the **Help** menu on the **Menu** bar.
2. The Configuration Manager will prompt you with a form to collect some or all the configuration, initialization, log, and dump files. The files will be organized into a directory and compressed as a ZIP file; you can share these files individually with MOS.

Note: Some information in your LEC Server configuration may be considered confidential or restricted by your organization. Consult your organization's procedures for sharing sensitive information with vendors before sharing your diagnostic data with MOS. If your organization will not allow you to share the information that the Configuration Manager gathered, let MOS know and they can work with you to sanitize any sensitive information.

Running Live Energy Connect as a Process

Sometimes while developing, troubleshooting, or testing with an Oracle Utilities Live Energy Connect (LEC) Server configuration, you may not want to run the server as a Windows service you might need to frequently start and stop it. In situations like this, you can run the server as a Windows process.

To run LEC Server as a Windows process:

1. With the desired configuration alias chosen, select **Start Server Process** from the **Configuration Alias** panel. The LEC Server will start as a process and a console window will appear (for ease of monitoring multiple LEC Server processes on one machine). If the console window is closed, the server process stops. The **Running State** indicator updates to **Running** or **Standby** once the server finishes its start-up (depending on the server configuration).

2. To stop an LEC Server from running as a server process, select **Stop Server** from the **Configuration Alias** panel when the appropriate configuration alias is selected. Alternatively, you can close the console window that was opened when the server process was started.

Note: To run multiple LEC Server instances on the same machine (as Windows services or processes), each server instance needs a unique **App Name**, **Soap port**, and **MMS port**. These server parameters are specified in the **Server** tab of the **Properties panel**.

Reloading Macros

As mentioned on the [Live Energy Connect Concepts and Terminology](#) page when the Oracle Utilities Live Energy Connect (LEC) Server starts, it uses the configuration defined in the Configuration Manager and the PDI macros defined in the **DataflowMacros.pdi** file to instantiate a server configuration.

Sometimes, customers may need to update which macros are loaded into a server configuration if they are upgrading from an older version of LEC or if they were provided custom PDI macros to use in their configuration.

To reload macros:

1. Stop any running LEC Server instances on the machine. To do this, select **Stop** from the **Services** tab in the **Central panel** (for running services). Alternatively, select **Stop Server** from the **Configuration Alias** panel (for any running server processes).
2. Unload any batch files by selecting the batch files in the **Batch Files** tab of the **Properties panel** and click **Unload**.
3. Select **Reload macros** from the **File** menu in the **Menu Bar**. The Configuration Manager will automatically start the LEC Server as a process and reload the macro definitions in the **DataflowMacros.pdi** file, or any custom macros specified in the **User macros** field, and then stop itself.

Creating an ICCP Client

Use the following steps to create an ICCP client.

Step 1: Open an Empty Configuration

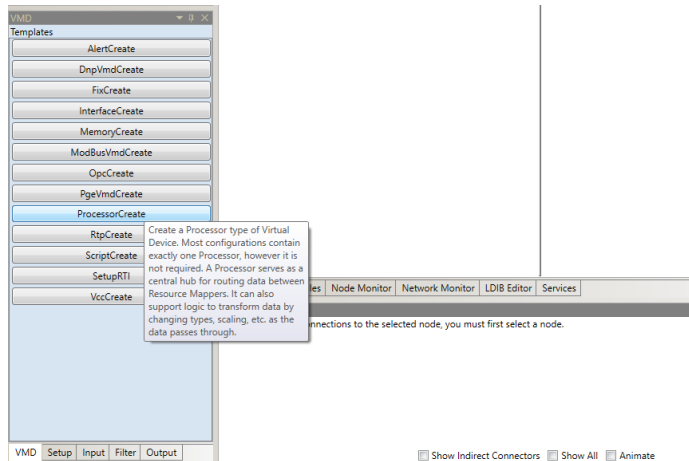
1. Start the Oracle Utilities Live Energy Connect (LEC) Configuration Manager as an Administrator.
2. Select **Import Configuration** from the **File** menu, and then select **EmptyTemplate.db** from the list of files in the **C:\ProgramData\LiveEnergyConnect** directory, and click **Open**.
3. The Configuration Manager displays a warning that any unsaved changes to the current configuration will be lost.
 - a. Click **Cancel** to save the current configuration in your desired location with an appropriate file name.
 - b. If you do not need to save any changes to the last open configuration, click **OK**.
4. The Configuration Manager may display a warning message indicating the **EmptyTemplate.db** configuration file contains PDI macros that are out of date. To update the macros, select **Reload Macros** from the **File** menu. A warning will appear saying the Configuration Manager needs to start the LEC Server. Click **OK**. The server will start up, reload any macros that need to be updated, and stop.
5. Once the **EmptyTemplate.db** file has been imported, you will be able to see an empty LEC Server configuration in the Configuration Manager. It will have a single VMD- a Configuration Manager VMD.

Note: Every new configuration has a Configuration Manager VMD.

Step 2: Create a Processor VMD

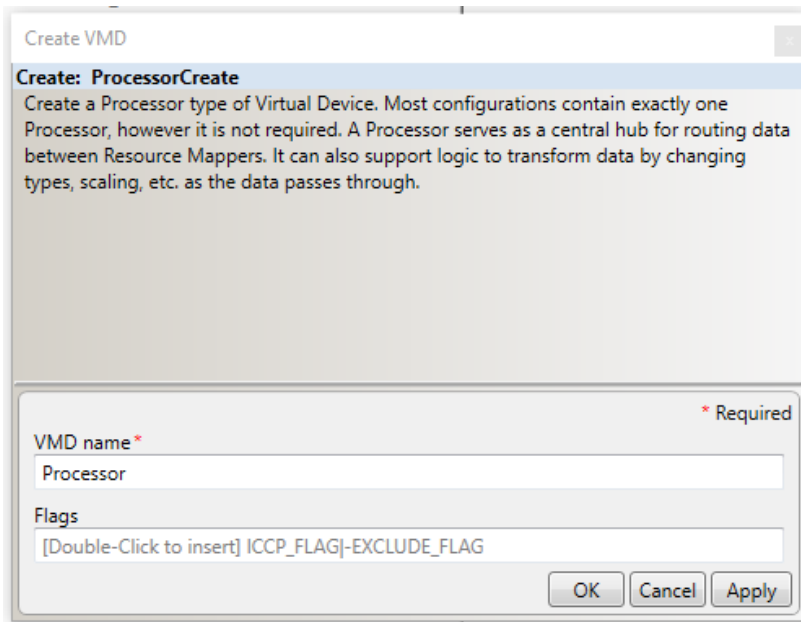
The first VMD you need to create is a Processor VMD. Any local VMD in a configuration can communicate with any other local VMD using a Processor VMD as a proxy. The Processor VMD acts as the hub for your dataflow.

1. Select **ProcessorCreate** in the **VMD** tab of the **Templates** panel.



2. In the **Create VMD** tool, specify **Processor** for the **VMD name** field.
3. There is no need to enable any flags in the **Flags** field.

Note: The **Flags** field allows you to enable or disable certain functions in a VMD. The **EXCLUDE_FLAG**, **ICCP_FLAG**, **PUSHALL_FLAG**, and **NOREAD_FLAG** flags are set by default. To remove a flag, click **Unset**. To set a flag, click **Set**.



4. Select **OK** to create the Processor VMD called **Processor**. Now there are two VMD's in the configuration: a Configuration Manager VMD and a Processor VMD.

Step 3: Create a VCC

In LEC, a VCC is a type of VMD. The term VCC is originates from the ICCP protocol.

Note: The term VMD is specific to the MMS protocol. All VCC's are VMD's but not all VMD's are VCC's. The VCC VMD created here will serve as the ICCP client that makes an ICCP association with the remote SCADA system.

To make an ICCP association, both parties need to know certain parameters and information about their own VCC and the other party's VCC. To learn more about setting up ICCP associations, see [ICCP Reference](#).

Typically, when two parties are setting up an ICCP association they share the necessary ICCP association information with each other using an Association Information Exchange Form (AIEF). Since we are setting up both sides of our example scenario, we can just use the example values that are listed in the following table. For information about each parameter, see [ICCP Reference](#).

ICCP Association and Network Connection Info	Company A (our LEC Server)	Company B (SCADA Server)
VCC Name	VCC_A	VCC_B
VCC Role	Client	Server
Association Role	Listener (makes inbound assoc.)	Associator (makes outbound assoc.)
Domain	domA	domB
Bilateral Table ID	1_0	2_0

ICCP Association and Network Connection Info	Company A (our LEC Server)	Company B (SCADA Server)
ICCP Version	1996, 8	1996, 8
Supported Features	111010000000	111010000000
Network Address	127.0.0.1:102	127.0.0.1:103
TSEL	00 01	00 02
SSEL	00 01	00 02
PSEL	00 00 00 01	00 00 00 02
AP Title	2 16 3826 86 67 67 65 49 73	2 16 3826 86 67 67 66 49 73
AE Qualifier	101	202

The table above provides ICCP association and network connection information for our example tutorial. This information will be used when creating VCC VMD's in the Configuration Manager.

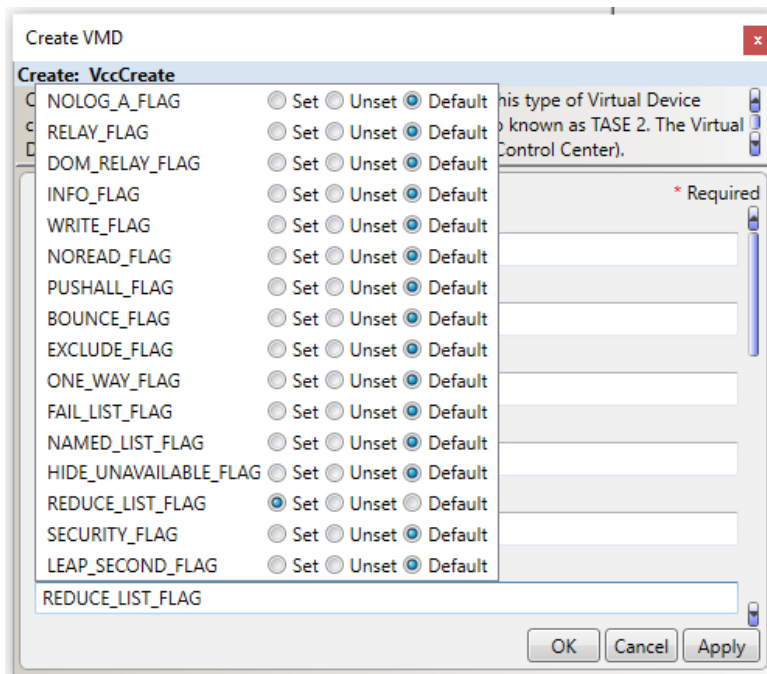
Create the Local VCC

1. In the **VMD** tab of the **Templates** panel, select the **VccCreate template**.
2. Specify the name of the VCC in the **My VMD** field as **VCC_A**.
3. Set **Assoc in** to **0** and **Assoc out** to **1**. This means the local VCC (VCC_A) will attempt to make an outbound ICCP association with its remote peer (as opposed to listening for an inbound association).
4. Set **Client role** to **1**.

5. Set **Server role** to **0**. In this example, the local VCC is just an ICCP client.

Note: A VCC can serve as an ICCP client and an ICCP Server (a dual-role VCC) simultaneously. A single VCC can also make associations with more than one other VCC. But for every role a VCC plays, it needs a dedicated association.

6. In the **Flags** field, set the **REDUCE_LIST_FLAG**. This flag lets the LEC Server adjust which points it includes in its transfer set requests if one or more of those points do not exist on the remote ICCP server.



Note: The string that populates the Flags field is determined by which flags are Set or Unset. A blank field uses the default setting for all flags. A flag that is explicitly set appears in this field as its name. A flag that is explicitly unset appears as its name with a minus sign ('-'), in front of it. Explicitly set or unset flag instructions are concatenated together with the pipe character ('|'). In this tutorial, this field appears as REDUCE_LIST_FLAG but another

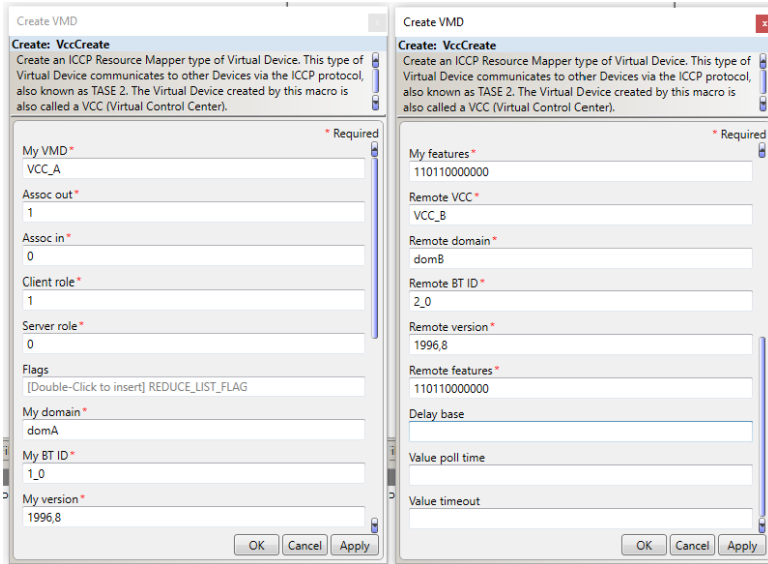
entry for this field might look something like **RELAY_**
FLAG|INFO_FLAG|EXCLUDE_FLAG.

7. Specify the domain of the server in the **My domain** as **domA**.
8. Type **1_0** in the **My BT ID** field. This field specifies the local VCC's bilateral table ID.
9. Type **1996,8** in the **My version** field.
10. In the **My features** field, type the 12-bit string, **110110000000**. In this example, the local VCC supports ICCP Blocks 1, 2, 4, and 5.

Note: While establishing an ICCP association, the client accesses a standard ICCP object called **Supported_Features**. This object is represented as a bit-string that specifies the ICCP Conformance Blocks supported in the server. To learn more about ICCP conformance blocks, see [ICCP Reference](#).

11. Type **VCC_B** in the **Remote VCC** field.
12. In the **Remote domain** field, type **domB**.
13. In the **Remote BT ID** field, type **2_0**.
14. Type **1996,8** in the **Remote version** field.
15. In the **Remote features** field, type **110110000000**.
16. Leave the **Delay base**, **Value poll time**, and **Value timeout** fields blank.

See the following image for a summary of what the fields in VccCreate tool should look like.



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

Specify the ICCP Association Parameters for the Local VCC

1. In the Configuration Manager, select the local VCC in the **Node Table** by clicking where it says **VCC_A**.
2. Then open the **VMD** tab of the **Properties** panel.
3. In the **VMD** tab of the **Properties** panel, click **My VMD**. A window that lets you specify the network connection and ICCP association parameters for VCC_A will appear :

Network Address	127.0.0.1:102
TSEL (ASCII)	00 01
TSEL	30 30 20 30 31
SSEL	00 01
PSEL	00 00 00 01
AP Title	2 16 3826 86 67 67 65 49 73
AE Qualifier	101
Supported Context	<input checked="" type="radio"/> International Standard (IS) <input type="radio"/> Draft (DIS)
Flags	
Secure ICCP	<input type="checkbox"/> Enable
Monitoring	<input checked="" type="checkbox"/> Enable
UCA	<input type="checkbox"/> Enable

OK Cancel Apply

Note: To fill in the information in this window, we will refer to the table of ICCP Association information that was included at the beginning of this section. To find out more about the meaning of each of these parameters, see [ICCP Node Templates in LCM](#).

4. Set the **Network Address** field to **127.0.0.1:102**.
5. Set the **TSEL (ASCII)** field to 00 01. This will cause the **TSEL** field to update to **30 30 20 30 31**.
6. Set the **SSEL** field to **00 01**.
7. Set the **PSEL** field to **00 00 00 01**.
8. Set the **AP Title** field to **2 16 3826 86 67 67 65 49 73**.
9. Set the **AE Qualifier** field to **101**.
10. Leave the **Supported Context**, **Secure ICCP**, **Monitoring**, and **UCA** fields as their default values.
11. Click **Apply**.

Specify the ICCP Association Parameters for the Remote VCC

1. In the Configuration Manager, select the local VCC in the Node Table by clicking where it says VCC_A.
2. Then open the **VMD** tab of the **Properties** panel.

3. In the **VMD** tab of the **Properties** panel, click **Remote VCC**. A window that lets you specify the network connection and ICCP association parameters for VCC_B will appear :

Network Address	127.0.0.1:103
TSEL (ASCII)	00 02
TSEL	30 30 20 30 32
SSEL	00 02
PSEL	00 00 00 02
AP Title	2 16 3826 86 67 67 66 49 73
AE Qualifier	202
Supported Context	<input checked="" type="radio"/> International Standard (IS) <input type="radio"/> Draft (DIS)
Flags	
Secure ICCP	<input type="checkbox"/> Enable
Monitoring	<input checked="" type="checkbox"/> Enable
UCA	<input type="checkbox"/> Enable

4. Set the **Network Address** field to **127.0.0.1:103**.
5. Set the **TSEL (ASCII)** field to **00 02**. This will cause the **TSEL** field to update to **30 30 20 30 32**.
6. Set the **SSEL** field to **00 02**.
7. Set the **PSEL** field to **00 00 00 02**.
8. Set the **AP Title** field to **2 16 3826 86 67 67 66 49 73**.
9. Set the **AE Qualifier** field to **202**.
10. Leave the **Supported Context**, **Secure ICCP**, **Monitoring**, and **UCA** fields as their default values.
11. Click **Apply**.

Step 4: Create a Script VMD

A Script VMD represents an interface to a Python script running in LEC's embedded real-time Python interpreter. You can create your own scripts or use one of the many scripts that ship with the LEC installer.

In this tutorial, your Script VMD will be an interface to a Python script called PointLogger.py. This script is part of your LEC installation and is located in the

C:\ProgramData\LiveEnergyConnect\Scripts directory. This Script VMD will get point values from the Processor write the values of the points to a specified log file as they come in.

Note: Although, this particular script's functionality is quite simple, Script VMD's provide the ability to perform complex filtering. They can be used to quickly create interfaces to non-standard, external applications.

Create the configurations script VMD

1. Navigate to the **Templates** panel and select the **VMD** tab.
2. Select the **ScriptCreate** template which will launch the **ScriptCreate** tool.
3. Enter **Logger** in the **My VMD** field.
4. In the **Flags** field, specify **ICCP_FLAG**. This allows the Script VMD to know about the standard ICCP datatypes.
5. Leave the Script file field blank.

Note: The **Script file** field is not where you specify the script associated with a Script VMD. This field instead allows you to specify another script that controls (enables or disables) this Script VMD. The script that Script VMD uses is specified later with a SetupScript node within the Script VMD.

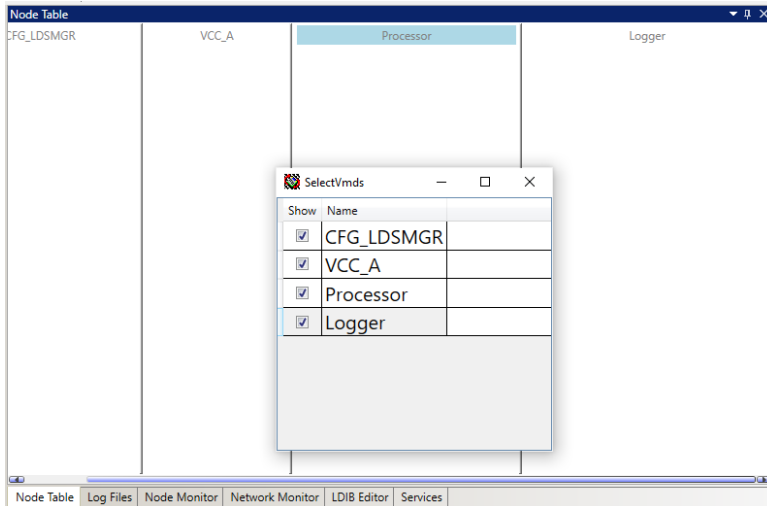
6. Leave the **UseCommonThread**, **Python instance**, and **Computer ID** fields blank.
7. Click **OK**.

Step 5: Arrange the VMD's in the Node Monitor

Before creating the nodes within each VMD and connectors between those nodes, it is helpful to arrange the VMD's in the Configuration Manager so that arrangement is representative of the dataflow we are creating. Typically, in engineering diagrams, data flows from left (input) to right (output).

Arrange the VMDs

1. Right-click the **Processor VMD** right where it says **Processor**.
2. Select **VCC_A** and drag it to place between **CFG_LDSMGR** and **Processor VMD**'s.



3. This updates the order in which the VMD's are displayed in the **Node Table**.

Note: To hide a VMD, select the checkbox next to the name of the VMD to hide. The order of the VMD's only affects the order in which they are displayed in RCM's Node Table. It does not affect the dataflow at all.

4. Save the configuration. Select **File** then select **Save Configuration**.
5. After creating all the required VMD's, the next step is to create the nodes inside those VMD's.

Step 6: Create Nodes in the VCC

Points coming from the remote ICCP server enter the local VCC (VCC_A) as input nodes with the type PointFromIccp. This sample configuration uses a batch file to specify these input nodes. You will need to create one prototype node that is used (along with the batch file) to generate all of the PointFromIccp nodes in the local VCC.

There are also a couple of setup nodes that are needed in the VCC_A VMD: two nodes to control the ICCP association and two nodes to control and define a Data Set Transfer Set.

Create the PointFromIccp Prototype Node

1. Select the VMD called **VCC_A**.
2. Navigate to the **Input** tab of the **Templates** panel. The list of possible input node templates used with the type of selected VMD is displayed.
3. Select **PointFromIccp** from the list of node templates.
4. In the **Point label** field, type **#clientPoints**.

Note: A point label that contains a string starting with the pound character ('#') signifies that the node is a prototype node. In one of the configuration's batch files there will be a table that contains a column (called the driven column) which matches the string starting with the pound character ('#').

5. In the **Type** field, enter **<type>**.
6. In the **Remote VCC** field, enter **VCC_B**.
7. Leave the **Remote domain** field blank.
8. In the **TS num** field, enter **0**. This will make the ICCP client (VCC_A) ask the remote ICCP server to organize the points it cares about into the Data Set Transfer Set called **TransferSet_0**.
9. In the **Combine group** field, enter **PointsFromICCP**.

10. Select **OK**.

The screenshot shows a 'Create Node' dialog box with the following fields and values:

- Create:** PointFromIccp
- Indication Point Data From ICCP Peer**
- Point label ***: #clientPoints
- Type ***: <type>
- Remote VCC ***: VCC_B
- Remote domain**: [Double-Click to insert] domB
- TS num ***: 0
- Combine group**: PointsFromICCP
- Period**: (empty)

Buttons at the bottom: OK, Cancel, Apply.

VCC_A acts as an ICCP client so it makes an outbound ICCP association. A VccAssocOutControl setup node and a related Constant input node is used to specify this outbound association. The Constant node is used to specify an initial value for the VccAssocOutControl node at start-up.

Create the VccAssocOutControl Node

1. With **VCC_A** still selected, navigate to the **Setup** tab of the **Templates** panel.
2. In the **Point label** field, enter the node name **assocOutCtrl**.
3. In the **Remote Vcc** field, enter **VCC_B**.
4. In the **Init state var** field, type **assocOutCtrl_init**. This is the node label that will be used for the Constant node that defines the initial state of this VccAssocOutControl node. This field needs to match the node label of the Constant node exactly.
5. Leave the **Association flags** and **Stale time** fields blank.

6. In the **Combine group** field, enter **AssociationControl**.
7. Click **OK**.

Note: The **Combine group** field is used to organize different nodes into groups. Nodes with the same value for the Combine group field will be displayed together underneath a combine group label in the Node Table and Node Monitor tabs.

Create the Constant Node that Will Initialize the Value of the VccAssocOutControl Node

1. With the **VCC_A** still selected, navigate to **Input** tab of the **Templates** panel. This will list all possible types of input nodes that can be used with the type of the selected VMD.
2. Select the **Constant** option from the list of node templates in the **Input** tab.
3. In the **Point label** field, enter the node name **assocOutCtrl_init**.

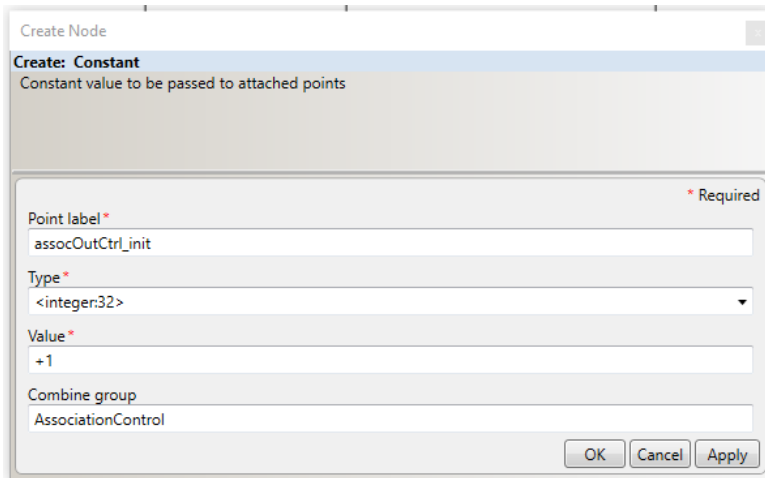
Note: Two (non-prototype) nodes that live in the same VMD and the same point domain within that VMD cannot have the same point label. However, one of these (non-prototype) nodes point labels can contain a substring that is another node's entire point label.

4. In the **Type** field enter **<integer:32>** which in the LEC Server's syntax for MMS, defines a 32-bit integer.
5. In the **Value** field enter **+1**. This will be the initial value that the node **assocOutCtrl** uses at start-up.

Note: In the LEC Server, an ICCP association variable can be in one of 3 states represented by an integer value: disabled (+0), enabled (+1), or associated (+2).

6. In the **Combine group** field enter **AssociationControl**.

7. Click **OK**.



Create Node

Create: Constant
Constant value to be passed to attached points

Point label* * Required
assocOutCtrl_init

Type*
<integer32>

Value*
+1

Combine group
AssociationControl

OK Cancel Apply

Note: The Configuration Manager automatically creates an Implicit Edge connector from the Constant node created in the last step to the node called assocOutCtrl which references this Constant node.

VCC_A will act as an ICCP client. When ICCP Block 2 is supported by both ICCP peers, an ICCP client can define Data Set Transfer Sets. These Data Set Transfer Sets tell the server which points the client wants to get information about and how it wants that information delivered.

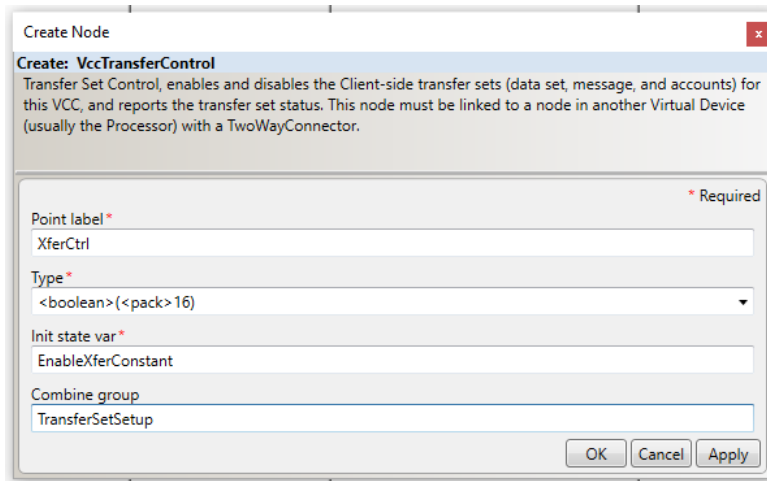
Note: Data Set Transfer Sets are a type of ICCP object.

To configure a Data Set Transfer Set in the LEC Configuration Manager, we will use a VccTransferControl setup node, a related Constant input node, and a DsTransferSet setup node.

Create the VccTransferControl Node

1. With the **VCC_A** still selected, open the **Setup** tab in the **Templates** Panel.
2. Select **VccTransferControl** from the list of node templates in the **Setup** tab.
3. In the **Point label** field, enter the node name **XferCtrl**.
4. In the **Type** field, enter **<boolean>(<pack>16)**, which in the LEC Server's syntax for MMS defines a structure made up of 16 bits.

5. In the **Init state var** field, enter **EnableXferConstant**. This will be the node label used for the Constant node that defines the initial state of this VccTransferControl node. Again, this field needs to match the node label of the related Constant node exactly.
6. In the **Combine group** field **TransferSetSetup**.
7. Click **OK**.



Create the Constant Node That Will Initialize the Value of the VccTransferControl Node

1. With **VCC_A** still selected, open the **Input** tab in the **Templates** Panel.
2. Select the **Constant** option from the list of node templates in the **Input** tab.
3. In the **Point label** field enter the node name **EnableXferConstant**.
4. In the **Type** field, enter **<boolean>(<pack>16)**.
5. In the **Value** field, enter **[T,T,T,T,T,T,T,T,T,T,T,T,T,T,T]**. This will be the initial value that the node called **XferCtrl** uses at start-up.

Note: By making all 16 bits of the VccTransferControl node have a value of **True**, you are enabling VCC_A to define up to 16 Data Set Transfer Sets. After a node with the type **<boolean>(<pack>n)** has been created, you can enable or disable a particular bit by clicking the text entry area for **Value** field. A pop-

up with radio buttons for each bit will appear and allow to select which bits in the object should be set to **True** and **False**.

6. In the **Combine group** field, enter **TransferSetSetup**.
7. Click **OK**.

Create Node

Create: **Constant**
Constant value to be passed to attached points

Point label* * Required
EnableXferConstant

Type*
<boolean>(<pack>16)

Value*
[[T,T,T,T,T,T,T,T,T,T,T,T,T,T,T]]

Combine group
TransferSetSetup

OK Cancel Apply

Create the DsTransferSet Node

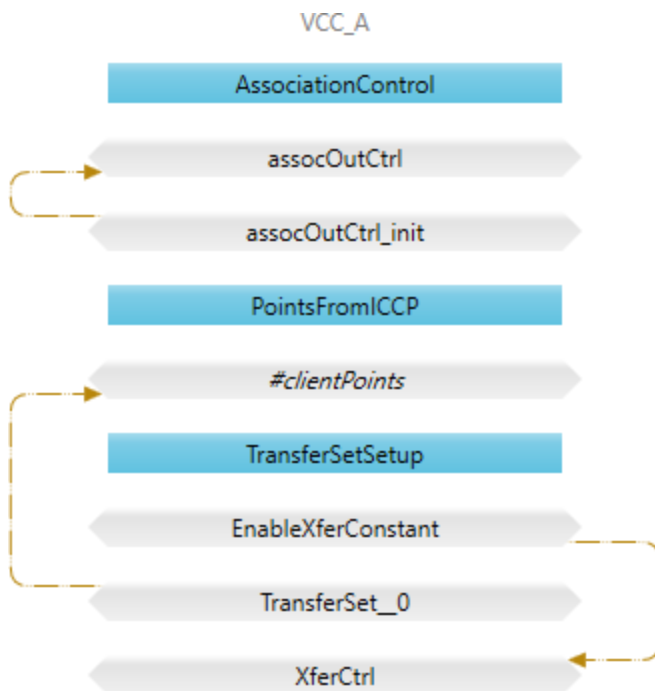
1. With the **VCC_A** still selected, open the **Setup** tab in the **Templates** Panel.
2. Select **DsTransferSet** from the list of node templates in the **Setup** tab.
3. In the **TS num** field, enter the node name **0** so that the whole node label will be **TransferSet_0**.
4. In the **Remote domain** field, enter **domB**.
5. In the **Remote VCC** field, enter **VCC_B**.
6. Leave the **Start time** and **Start delay** fields blank.
7. In the **IntervalCheck** field, enter **3**.
8. Leave the **Start time** and **Start delay** fields blank.
9. In the **IntegrityCheck** field, enter **0**.
10. In the **IntervalTimeOut** field, enter **0**.
11. In the **IntegrityTimeOut** field, enter **1**.
12. In the **ObjectChange** field, enter **1**.
13. In the **OperatorRequest** field, enter **1**.

- In the **RBE** field, enter **1**.
- Leave the **IntervalDelay**, **AllChangesReported**, **Critical**, **CircumventSiemensBug**, and **Do read** fields blank.

Note: For more information about each field in the DsTransferSet node template, see [ICCP Reference](#).

- In the **Combine group** field, enter **TransferSetSetup**.
- Click **OK**.

At this point in the tutorial, there should be five different nodes in VCC_A.



Notice that the nodes are organized into three different combine groups each with its own label: **AssociationControl**, **PointsFromICCP**, and **TransferSetSetup**. Also, notice that the Configuration Manager has automatically built three implicit edge connectors (brown arrows) because of the relationships that were defined between some of the nodes.

Step 7: Create Nodes in the Processor VMD

Any VMD in an LEC Server configuration can communicate with any other VMD in the configuration using a Processor VMD as a proxy. A Processor VMD will also typically contain nodes used for filtering data or for monitoring and controlling associations.

The `IntermediatePointMonitor` node is one of the most used types of nodes in a Processor VMD. Although it is a filter node, the `IntermediatePointMonitor` node is used to read (and write) the value of a point at a particular step in its dataflow through the LEC server.

In this example ICCP client configuration, you will create `IntermediatePointMonitor` nodes and use them to get the data received by the `VCC_A` VMD into the Processor VMD.

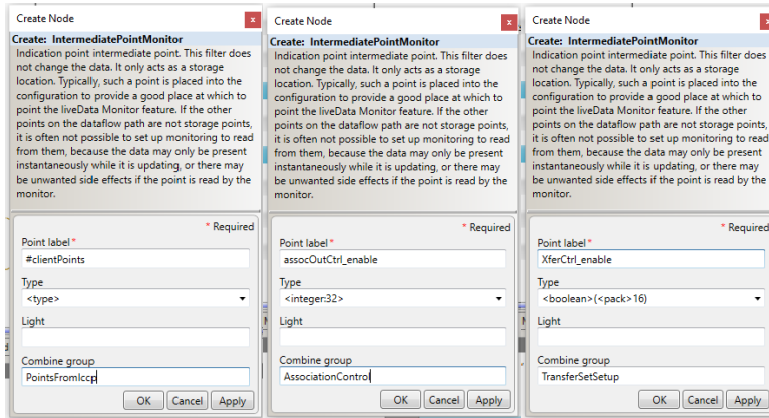
You will also create `IntermediatePointMonitor` nodes to allow the Configuration Manager and the LEC Server to read and write to the nodes in the `VCC_A` that are control the outbound association and Data Set Transfer Set control.

Create Three `IntermediatePointMonitor` Nodes in the Processor VMD

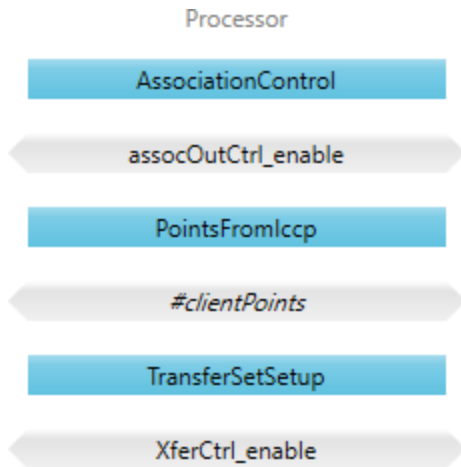
1. Select the **VMD Processor**.
2. Open the **Filter** tab in the **Templates** panel.
3. Select **IntermediatePointMonitor** from the list of node templates in the **Filter** tab.
4. In the **Point label** field, enter the node name **#clientPoints**.
5. In the **Type** field, type **<type>**.
6. Leave the **Light** field blank.
7. In the **Combine group** field, enter **PointsFromIccp**.
8. Select **OK**.
9. Repeat steps 1 -8 to create two more `IntermediatePointMonitor` nodes in the Processor VMD: one called **assocOutCtrl_enable** and one called **XferCtrl_enable**. The values for all three `IntermediatePointMonitor` nodes in the VMD are specified in the table below.

Template Node Field	#clientPoints	assocOutCtrl_enable	XferCtrl_enable
Point label	#clientPoints	assocOutCtrl_enable	XferCtrl_enable
Type	<type>	<integer:32>	<Boolean> (<pack>16)
Combine Group	PointsFromlccp	AssociationControl	TransferSetSetup

10. While creating each node, check that each matches the relevant template node in the figure below to ensure that there are no typos.



These IntermediatePointMonitor nodes are the only nodes that need to be created in the Processor VMD. At this point in the tutorial, there should be three different nodes in the Processor VMD as is shown in the figure below.



Step 8: Create Nodes in the Script VMD

The Script VMD called **Logger** will need a setup node and an output node. The setup node, a SetupScript node, will specify information about which Python script to use in the VMD. The output node, a PointToScript node will define the output behavior of the Script VMD based on the methods in the Python script. In this example, the output behavior of the Logger Script VMD will be to write changes in value to a CSV file.

Create a SetupScript Node in the Logger Script VMD

1. Navigate to **Setup** tab of the **Templates** panel.
2. Select the **SetupScript** template to display the **Create Node** form.
3. Enter **PointLogger** in the **Script label** field.
4. Enter **PointLogger.py** in the **Script file** field.
5. Enter **C:\\ProgramData\\LiveEnergyConnect\\Logs** in the **Arg1** field. Make sure the file path is in double quotes and contains double backslashes.
6. Type **Setup** in the **Combine Group** field.

7. Click **OK**.

Node

Properties: SetupScript

Setup of Script Agent, makes an instance of a scriptList RTX

* Required

Script label*
PointLogger

Script file*
PointLogger.py

Period

Type

RPC1

RPC2

RPC3

Arg1
"C:\\ProgramData\\LiveEnergyConnect\\Logs"

Arg2

Arg3

Cancel Apply

Server VMD Node Connector Batch Files

Create a PointToScript Node in the Logger Script VMD

1. Navigate to **Output** tab of the **Templates** panel.
2. Select **PointToScript** from the list of templates.
3. Type **#clientPoints** in the **Point label** field.
4. Type **<type>** in the **Type** field.
5. Type **#clientPoints** in the **Tag** field.
6. Type **PointLogger** in the **Script label** field.

7. Leave the **Python instance** field blank, and type **PointsToScript** in the **Combine group** field.
8. Click **OK** to finish this step.

Create Node

Create: **PointToScript**
Data to Script Agent by means of writing to the script, makes an instance of a scriptListElement RTX

* Required

Point label *
#clientPoints

Type *
<type>

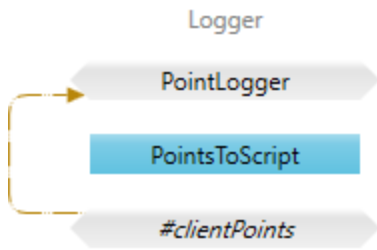
Tag
#clientPoints

Script label *
PointLogger

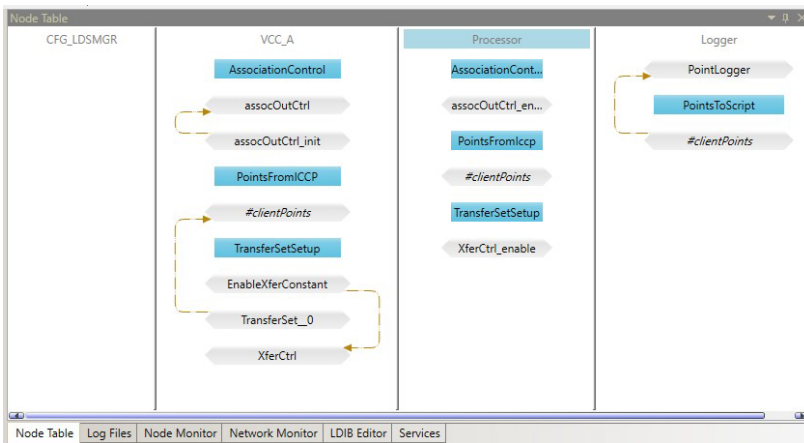
Combine group
PointsToScript

OK Cancel Apply

These two nodes are the only nodes you need to create in the Script VMD. At this point in the tutorial, there should be two different nodes in the Script VMD called **Logger** as shown below.



All nodes for our example LEC Server configuration have been created. If everything was specified correctly, the view of the configuration from the **Node Table** tab of the **Central** panel should look like the one shown in the figure below.



The last step in building the configuration is to connect the nodes.

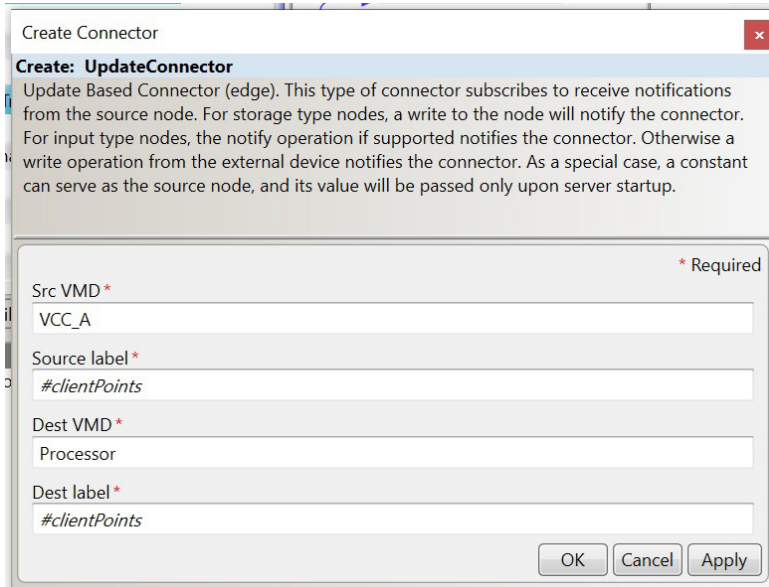
Step 9: Connect the Nodes

This LEC Server configuration requires Update connectors and Two Way connectors between some of the nodes.

Create Connectors Between Nodes in the Configuration

1. Click the **Create Update Connector** icon (the blue arrow) from the menu.
2. The **Create Connector** tool window will open.
3. Type **VCC_A** in the **Src VMD** field.
4. Type **#clientPoints** in the **Source label** field.
5. Type **Processor** in the **Dest VMD** field.

6. Type **#clientPoints** in the **Dest label** field.



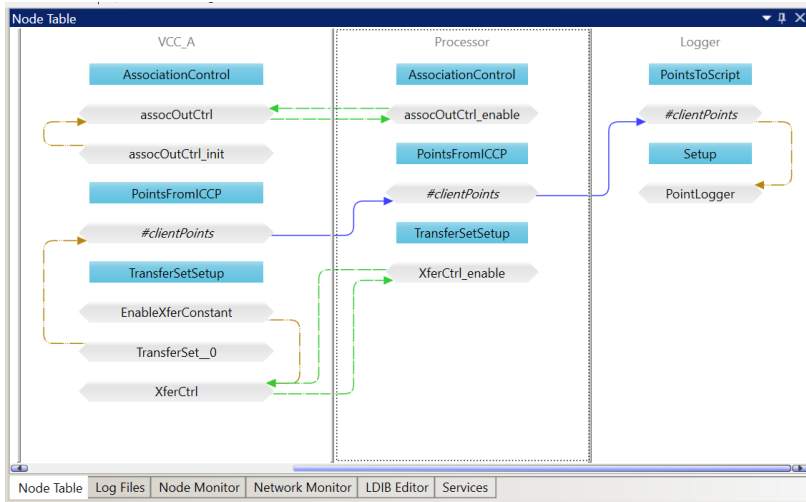
7. Click **OK**.

8. Use the Create Connector tool and the values in the following below to create the remaining three connectors in this example server configuration.

The following table shows the values used to create each Update Connector and Two-way Connector in this example server configuration.

Connector Type	Source VMD	Source Node	Destination VMD	Destination Node
Update Connector	VCC_A	#clientPoints	Processor	#clientPoints
Update Connector	Processor	#clientPoints	Logger	#clientPoints
Two-Way Connector	VCC_A	XferCtrl	Processor	XferCtrl_enable
Two-Way Connector	VCC_A	assocOutCtrl	Processor	assocOutCtrl_enable

Once all four of these connectors have been created, the view of your LEC Server configuration from the **Node Table** tab should look like the one shown in the figure below.



Step 10: Validate the Configuration

Now that the example configuration is completed, let's validate that it works just like the provided example configuration.

Load the Batch File

1. Navigate to the **Batch Files** tab of the **Properties** panel and select **Load Share**.
2. When prompted select the example batch file called **lccpPoints.csv** from the **C:\ProgramData\LiveEnergyConnect\Config** directory.
3. Confirm that the prototype nodes in the configuration generate the nodes defined in the batch file as expected.

Note: If loading the batch file doesn't generate any nodes or if you receive an error message from the Configuration Manager, one or more field values in the relevant prototype node does not match the column values in the batch file.

Start the LEC Server as a Server Process

1. Confirm that the **Soap port** and **Mms port** are specified as **8089** and **103** respectively in the **Server** tab of the **Properties** panel. If not, specify the field

values and click **Apply**.

2. Click the **Start Server Process** button in the **Configuration Alias** panel.
3. Confirm that the server starts by looking at the **Running Status** in the **Configuration Alias** panel.
4. Confirm that the server is attempting to make an outbound ICCP association with the remote ICCP server by looking at the **Network Monitor** tab of the **Central** pane. The association between VCC_A and VCC_B should have a status that reads **Attempting**.

If you are working through this tutorial you should have already created the configuration alias called ScadaSim from the earlier tutorial section of the guide called [A Quick Configuration Manager Tutorial](#).

If you skipped that section or if you just no longer have the example server configuration loaded for the ScadaSim configuration alias, then refer to that tutorial to prepare the SCADA server simulator configuration.

Test Against the Simulated SCADA Server Example Configuration

1. Switch to the Configuration Alias **ScadaSim** using the drop-down selector in the **Configuration Alias** panel.
2. Confirm that the **IccpSimPoints.csv** batch file is loaded by looking at the **Batch Files** tab in the **Properties** panel. If it is not loaded, then load it is using **Load Share**.
3. Click **Start Server Process** in the **Configuration Alias** panel.
4. Confirm that the server starts by looking at the **Running Status** in the **Configuration Alias** panel.
5. Confirm that the two running server instances establish an ICCP association using the **Network Monitor** tab of the **Central** pane. The association between VCC_A and VCC_B should have a status that reads **Connected**.
6. Switch back to the configuration **alias cfg**.
7. Confirm that point values are being updated in the Processor VMD of this server instance. You can verify this using the **Node Monitor** tab in the **Central** pane.
8. Finally, confirm that the embedded Python script used in the Logger VMD is logging point values in CSV file in the specified directory (specified in the **Arg1** field of the VMD's SetupScript node).
9. Save the configuration. Click **Save As** from the **File** menu.

You have now successfully created your first LEC Server configuration from scratch.

If you are having trouble validating this LEC Server configuration, try the following:

- Confirm there are no missing nodes or connectors.
- Confirm there are no typos in the values used for the various parameters of the VMD's, nodes, and connectors.
- Confirm there are not two running instances of the server configured to use the same ports for SOAP and MMS communication.
- Confirm that you are not running two instances of LEC Configuration Manager simultaneously.

For any further issues, contact [My Oracle Support](#).

ICCP Reference

The following sections describe how Inter-Control Center Communications are used with Oracle Utilities Live Energy Connect.

ICCP Indication Point Types in Oracle Utilities Live Energy Connect

The following is a list of the possible Oracle Utilities Live Energy Connect (LEC) Server types that ICCP points can have and the each type's structure in the Oracle Utilities Live Energy server. The bold typeface in the list below is the syntax used to specify each type in the LEC server (e.g. the Type field of a PointFromIccp node might be specified as **Data_RealQ**).

Data_Real

floating-point: {format-width 24, exponent-width 8}

Data_State

bit-string:

{

```
State_hi[0],
State_lo[1],
Validity_hi[2],
Validity_lo[3],
CurrentSource_hi[4],
CurrentSource_lo[5],
NormalValue[6],
TimeStampQuality[7]
}
```

Data_Discrete

```
integer: {width 32}
```

Data_Flags

```
bit-string:
```

```
{
unused[0],
unused[1],
Validity_hi[2],
Validity_lo[3],
CurrentSource_hi[4],
CurrentSource_lo[5],
NormalValue[6],
TimeStampQuality[7]
}
```

Data_TimeStamp

```
integer: {width 32}
```

Note: The integer value represents seconds since Unix epoch.

COV_Counter

Unsigned: {width 16}

The following complex types are used in transferring IndicationPoint object values:

Data_RealQ STRUCTURE

```
{  
COMPONENT Value Data_Real,  
COMPONENT Flags Data_Flags  
}
```

Data_StateQ

Data_State

Data_DiscreteQ STRUCTURE

```
{  
COMPONENT Value Data_Discrete,  
COMPONENT Flags Data_Flags  
}
```

Data_RealQTimeTag STRUCTURE

```
{  
COMPONENT Value Data_Real,  
COMPONENT TimeStamp Data_TimeStamp,  
COMPONENT Flags Data_Flags  
}
```

Data_StateQTimeTag STRUCTURE

```
{  
COMPONENT TimeStamp Data_TimeStamp,  
COMPONENT Flags Data_State  
}
```

Data_DiscreteQTimeTag STRUCTURE

```
{  
COMPONENT Value Data_Discrete,  
COMPONENT TimeStamp Data_TimeStamp,  
COMPONENT Flags Data_Flags  
}
```

Data_RealExtended STRUCTURE

```
{  
COMPONENT Value Data_Real,  
COMPONENT TimeStamp Data_TimeStamp,  
COMPONENT Flags Data_Flags,  
COMPONENT COV COVCounter  
}
```

Data_StateExtended STRUCTURE

```
{  
COMPONENT TimeStamp Data_TimeStamp,  
COMPONENT Flags Data_State,  
COMPONENT COV COVCounter  
}
```

Data_DiscreteExtended STRUCTURE

```
{  
COMPONENT Value Data_Discrete,
```



```

COMPONENT TimeStamp Data_TimeStamp,
COMPONENT Flags Data_Flags,
COMPONENT COV COVCounter
}

```

Note: Data_RealQTimeTagExtended uses `TIMESTAMP_EXTENDED` instead of `Data_TimeStamp`. The `TimeTagExtended` types have a millisecond time field in addition to the timestamp (and no COV field). `TimeTagExtended` types were added in the 2000,8 version of ICCP.

IndicationPointConfig STRUCTURE

```

{
COMPONENT PointType integer { width 8, range 0 .. 2 },
COMPONENT QualityClass integer { width 8, range 0 .. 1 },
COMPONENT NormalSource integer { width 8, range 0 .. 3 },
COMPONENT TimeStampClass integer { width 8, range 0 .. 1 },
COMPONENT COVClass integer { width 8, range 0 .. 1 },
}

```

ICCPQualityCodes

The ICCP protocol provides quality codes to represent data quality for a point. If used, these are transferred with each data item.

The quality codes for a data item are derived from the SCADA system's reliability information. The mapping of a SCADA system's data reliability information into the ICCP quality codes is a local implementation issue and can vary across organizations.

The categories of data item quality information are:

- **Data Item Status or Validity**

- Valid (Good)
- Invalid (Bad)
- Held
- Suspect

- **Data Item Source:**

- Telemetered
- Calculated
- Estimated
- Manual

- **Data Item State:**

- Normal
- Abnormal (off-normal)

Oracle Utilities Live Energy Connect Configuration Manager Node Templates for ICCP

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

ICCP Setup Node Templates

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

- VccAssocInControl
- VccAssocOutControl
- VccTransferControl
- VerifyAssociation

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

ICCP Conformance Blocks

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

Block	Definition	Related
Block 1 Periodic System	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)
Block 2 Extended Data Set Condition Monitoring	Provides report by exception capability for the data types that block 1 is able to transfer periodically.	PointFromIccp (Input Node for Blocks 1 and 2) PointToIccp (Output Node for Blocks 1 and 2)
Block 4 Information Messages	Provides support for simple text and binary.	ImTransferSet (Setup Node for Block 4) MessageFromIccp (Input Node for Block 4)
Block 5 Device Control	Provides device control requests: on/off, trip/close, raise/lower node and digital set points. Includes mechanisms for interlocked controls and select-beforeoperate	ControlFromIccp (Input for Block 5) ControlToIccp (Output for Block 5)
Block 8	Provides scheduling,	GetOutageFromIccp (Input Node

Block	Definition	Related
Additional User Objects	<p>accounting, and outage and plant information.</p> <p>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</p>	<p>for Block 8)</p> <p>GetTAQueryFromIccp</p> <p>SendOutageToIccp (Output Node for Block 8)</p> <p>SendTAQueryToIccp (Output Node for Block 8)</p> <p>TANoSegPeriodicFromIccp (Input Node for Block 8)</p> <p>TANoSegPeriodicToIccp (Output Node for Block 8)</p> <p>TAServerTransferSetFromIccp (Input Node for Block 8)</p>

ICCP Node Templates in LCM

VccCreate (VMD Resource Mapper Template)

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

Parameter	Definition	Default
Assoc out	Required. Specifies whether this VCC is to have an outbound association. 1 enables an outbound association; 0 does not.	Yes, unless the scope is VCC-wide.
Assoc in	Required. Specifies whether this VCC is to have an inbound association. 1 enables an inbound association; 0 does not. <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p>Note: A VCC can have both an inbound and an outbound association.</p> </div>	No, if the data type is COMMAND
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	<p>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.</p> <p>The flags argument is expressed as a series of flag keywords separated by the vertical bar (' ') character in the Flags field.</p> <p>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.</p>	<p>The PUSHALL_FLAG, EXCLUDE_FLAG, FAIL_LIST_FLAG, and NAMED_LIST_FLAG flags are set by default.</p> <p>All other flags are not set by default unless both the Assoc out and</p>

Parameter	Definition	Default
	To remove a flag, unset it in the form or precede it with a minus sign in the Flags field.	Assoc in parameters are set to ' 1 ', then the ONE_WAY_FLAG is set by default. The EXCLUDE_FLAG is set by default.
	NOLOG_A_FLAG: If set, directs LEC Server not to generate log messages for variable access failures.	Not Set
	RELAY_FLAG: If set, directs LEC 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 LEC 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 LEC 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 LEC	Not Set

Parameter	Definition	Default
	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 LEC Server not to perform an initial read of all push-list variables, directing LEC Server to refresh the variable's internal state only as writes come in.	Not Set
	PUSHALL_FLAG: If set, directs LEC Server not to check push-list variables for changes in value, directing LEC Server to push a variable whenever a write comes in regardless of the value.	Not Set
	BOUNCE_FLAG: If set, directs LEC 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
	EXCLUDE_FLAG: If set, directs LEC 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 LEC Server to force outbound requests to use an outbound association.	Not Set
	FAIL_LIST_FLAG: If set, directs LEC Server to prevent the creation of a variable list if any of its variables are non-existent.	Not Set

Parameter	Definition	Default
	NAMED_LIST_FLAG: If set, directs LEC 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 LEC 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 LEC Server to perform the secure ICCP procedures (certificate checking) at the ACSE level.	Not Set
	LEAP_SECOND_FLAG: If set, directs LEC Server to adjust timestamps according to the official schedule of leap seconds (as found in the srvxnt.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

Parameter	Definition	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 LEC Server is to wait for all variables to have a value before allowing an	0 seconds

Parameter	Definition	Default
	association. A value of 0 disables the feature.	

ControlFromIccp (Input Node Template)

Takes device control from the ICCP peer.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within LEC Server. This name will be displayed throughout LEC 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 LEC 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	<p>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.</p> <p>The flags argument is expressed as a series of flag keywords separated by the vertical bar (' ') character in the Flags field.</p> <p>If you prefer to use a form, then click</p>	No, if the data type is COMMAND

Parameter	Definition	Required
	<p>in the Flags field to view the form that allows you to set and unset flags using radio buttons.</p> <p>To remove a flag, unset it in the form or precede it with a minus sign in the Flags field.</p> <p>REAL: Sets the control point to type REAL. Default: If not set, the default is type COMMAND.</p> <p>DISCRETE: Sets the control point to type DISCRETE. Default: If not set, the default is type COMMAND.</p>	
	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

ControlTolccp (Output Node Template)

Transfers device control to the ICCP peer.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the	Yes

Parameter	Definition	Required
	<p>data within LEC Server. This name will be displayed throughout LEC 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 LEC Configuration Manager, for example, when connecting this node to another node.</p>	
Remote VCC	<p>Specifies the common name of the remote VCC. Use the name of the server VCC if the client and server are different.</p>	Yes
Remote Domain	<p>Specifies the remote domain name for server-side data. Leave this parameter blank if the scope is VCC-wide.</p>	Yes, unless the scope is VCC-wide.
Flags	<p>This parameter is a set of Boolean (on/off) options that allows you to select a data type and specify additional options.</p> <p>The flags argument is expressed as a series of flag keywords separated by the vertical bar (' ') character in the Flags field.</p> <p>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.</p> <p>To remove a flag, unset it in the form or precede it with a minus sign in the</p>	No, if the data type is COMMAND.

Parameter	Definition	Required
	<p>Flags field.</p> <p>REAL: Sets the control point to type REAL. Default: If not set, the default is type COMMAND.</p> <p>DISCRETE: Sets the control point to type DISCRETE. Default: If not set, the default is type COMMAND.</p>	
	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(SetupNodeTemplate)

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	Yes

Parameter	Definition	Required
	the remote server's VCC.	
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

Parameter	Definition	Required
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	No

Parameter	Definition	Required
	enabled.	
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
SiemensCompatibility	Indicates whether or not to circumvent the Siemens Transfer Set domain name bug. Specifying any value directs LEC Server is to circumvent the Siemens bug. No value indicates not to circumvent this bug.	No
Do read	Indicates whether or not LEC Server is to perform an initial read. 1 indicates to do an initial read of data values; 0 or no value indicates that LEC 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(InputNodeTemplate)

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

Parameter	Definition	Required
Point label	<p>Assigns a name that is for routing the data within LEC Server. This name will be displayed throughout LEC 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 LEC Configuration Manager. This specific label indicates the information requested about the outage.</p> <p>Specify one of the following:</p> <ul style="list-style-type: none"> ▪ 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 canceled if it was canceled. 	Yes
Type	<p>Specifies the data type for this outage object. Use the data type with the same name as the Point label:</p> <ul style="list-style-type: none"> ▪ DONewRevSched ▪ DOActual ▪ DOCancel 	Yes
Remote VCC	<p>Specifies the common name of the remote VCC. Use the server VCC name if the client and server are different.</p>	Yes

Parameter	Definition	Required
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(InputNodeTemplate)

Retrieves the Transfer Account Query from the ICCP peer.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within LEC Server. This name will be displayed throughout LEC 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 LEC 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(SetupNodeTemplate)

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(InputNodeTemplate)

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 LEC Server. This name will be displayed throughout LEC 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 LEC 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

Parameter	Definition	Required
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(OutputNodeTemplate)

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 LEC Server. This name will be displayed throughout LEC 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 LEC 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 Template)

Retrieves indication point data from the ICCP peer.

Parameter	Definition	Required
Point label	<p>Assigns a name that is for routing the data within LEC Server. This name will be displayed throughout LEC 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 LEC Configuration Manager, for example, when connecting this node to another node.</p>	Yes
Type	<p>The supported ICCP types are:</p> <ul style="list-style-type: none"> ▪ 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

Parameter	Definition	Required
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	<p>Places a limit (in seconds) on the time LEC 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 one requires more memory and will slow down performance.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p>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</p> </div>	No

Parameter	Definition	Required
	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">changes.</div> <p>If this parameter is undefined or set to 0, LEC Server will wait indefinitely.</p>	
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 Template)

Sends indication point data to the ICCP peer.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within LEC Server. This name will be displayed throughout LEC 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 LEC Configuration Manager, for example, when connecting this node to another node.	Yes
Type	<p>The supported ICCP types are:</p> <ul style="list-style-type: none"> ▪ Data_Discrete ▪ Data_Real ▪ Data_RealQ ▪ Data_RealQTimeTag 	Yes

Parameter	Definition	Required
	<ul style="list-style-type: none"> ▪ Data_RealExtended ▪ Data_RealQTimeTagExtended ▪ Data_State ▪ Data_StateQ ▪ Data_StateQTimeTag ▪ Data_StateExtended ▪ Data_StateQTimeTagExtended ▪ Data_Discrete ▪ Data_DiscreteQ ▪ Data_DiscreteQTimeTag ▪ Data_DiscreteExtended ▪ Data_DiscreteQTimeTagExtended 	
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 Template)

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

Parameter	Definition	Required
Point label	Assigns a name that is for routing the	Yes

Parameter	Definition	Required
	<p>data within LEC Server. This name will be displayed throughout LEC 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 LEC Configuration Manager.</p> <p>This specific label indicates information about the planned outage that LEC Server is to send to the ICCP peer.</p> <p>Specify one of the following:</p> <ul style="list-style-type: none"> ▪ 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. 	
Type	<p>Specifies the data type for this outage object. Use the data type with the same name as the Point label:</p> <ul style="list-style-type: none"> ▪ 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

Parameter	Definition	Required
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 Template)

Sends a query to the ICCP peer.

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within LEC Server. This name will be displayed throughout LEC 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 LEC 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 Template)

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	Yes

Parameter	Definition	Required
	<p>data within LEC Server. This name will be displayed throughout LEC 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 LEC Configuration Manager, for example, when connecting this node to another node.</p>	
Row Type	<p>Provides the MMS-DL (Manufacturing Messaging Service-Descriptive Language) specification of the type of data in a single matrix row.</p>	Yes
Num Floats	<p>Specifies the number of floats in a matrix row</p>	Yes
Num ints	<p>Specifies the number of integers in a matrix row.</p>	Yes
Num matrix ids	<p>Specifies the sum of the number of floats and integers in a matrix row.</p>	Yes
Max loc refs	<p>Specifies the maximum number of local refs that are allowed in the header structure</p>	Yes
Max rows	<p>Specifies the maximum number of matrix rows which may be passed using this point</p>	Yes
Ts num	<p>Specifies the number of the transfer set to which this point is assigned.</p>	Yes

Parameter	Definition	Required
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 Template)

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 LEC Server. This name will be displayed throughout LEC 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 LEC 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	Yes

Parameter	Definition	Required
	structure.	
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

TAServerTransferSetFromlccp (Input Node Template)

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

Parameter	Definition	Required
Point label	Assigns a name that is for routing the data within LEC Server. This name will be displayed throughout LEC 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 LEC Configuration Manager, for example, when connecting this node to another node.	Yes
Remote VCC	Specifies the common name of the	Yes

Parameter	Definition	Required
	remote server VCC.	

TaTransferSet (Setup Node Template)

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

Parameter	Definition	Required
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

Parameter	Definition	Required
DuringTheHour	<p>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.</p> <p>Often, this is referred to as "mid hour schedule changes."</p>	Yes
AfterTheHour	<p>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.</p> <p>Often, this is referred to as "after the hour actuals."</p>	Yes
ActualDataUpdate	<p>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.</p> <p>Often, this is referred to as "corrections to previous schedules."</p>	Yes
PastHours	<p>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.</p>	Yes
ObjectChange	<p>Indicates whether or not the server is to send a report when</p>	Yes

Parameter	Definition	Required
	any object in the transfer account changes. Specify 1 to direct the server to send a report when a change occurs; otherwise, specify 0.	
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 Template)

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 LEC Server. This name will	Yes

Parameter	Definition	Required
	<p>be displayed throughout LEC 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 LEC Configuration Manager, for example, when connecting this node to another node.</p>	
Remote VCC	<p>Specifies the common name of the remote VCC.</p>	Yes
Init state var	<p>Indicates whether or not LEC Server is to establish an association initially.</p> <p>Specify +1 if LEC Server is to enable the association initially; specify +0 if LEC Server is not to enable the association initially.</p>	Yes
Association Flags	<p>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.</p> <p>No</p> <p>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.</p> <p>To remove a flag, unset it in the form or precede it with a minus sign in the Flags field.</p>	No

Parameter	Definition	Required
	<p>RELAY_FLAG: If set, directs LEC 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.</p> <p>DOM_RELAY_FLAG: If set, directs LEC 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.</p> <p>INFO_FLAG: If set, directs LEC Configuration Manager to send out Information Reports instead of Event Notifications for pushed and polled variables and for enrolled MMS Read Events.</p> <p>BOUNCE_FLAG - If set, directs LEC 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.</p> <p>EN_OUT_FLAG - If set, directs LEC Configuration Manager to attempt to make an outbound association.</p> <p>SUPERCEDE_FLAG - If set and an associate request comes in while there is already an association, directs LEC Configuration Manager to abort the old association and accept the new one. If</p>	

Parameter	Definition	Required
	<p>not set, LEC Configuration Manager refuses the new association.</p> <p>ALLOW_MULTI_FLAG - If set and an associate request comes in while there is already an association, directs LEC 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.</p>	
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 Template)

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 LEC Server. This name will be displayed throughout LEC 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	Yes

Parameter	Definition	Required
	LEC Configuration Manager, for example, when connecting this node to another node.	
Remote VCC	Specifies the name of the remote VCC.	Yes
Init state var	<p>Indicates whether or not LEC Server is to establish an association initially.</p> <p>Specify +1 if LEC Server is to enable the association initially; specify +0 if LEC Server is not to enable the association initially.</p>	Yes
Association Flags	<p>Each of these flags disables or enables an option for handling associations controlled by this node. By default, none of these flags are set.</p> <p>The flags argument is expressed as a series of flag keywords separated by the vertical bar (' ') character in the Flags field.</p> <p>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.</p> <p>To remove a flag, unset it in the form or precede it with a minus sign in the Flags field.</p> <p>RELAY_FLAG: If set, directs LEC 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.</p>	No

Parameter	Definition	Required
	<p>DOM_RELAY_FLAG: If set, directs LEC 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.</p> <p>INFO_FLAG: If set, directs LEC Server to send out Information Reports instead of Event Notifications for pushed and polled variables and for enrolled MMS Read Events.</p> <p>WRITE_FLAG: If set, directs LEC Server to send out MMS Writes instead of Event Notifications for pushed and polled variables and for enrolled MMS Read Events.</p> <p>NOREAD_FLAG: If set, directs LEC Server to make all AA-specific PGE variables candidates for pushing (without the need for inclusion in a PUSH list).</p> <p>REPORT_FLAG: If set, directs LEC Server not to perform an initial read of all push- list variables, directing LEC Server to refresh the variable's internal state only as writes come in.</p> <p>BOUNCE_FLAG: If set, directs LEC 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.</p> <p>EN_OUT_FLAG: If set, directs LEC Server to attempt to make an outbound</p>	

Parameter	Definition	Required
	<p>association.</p> <p>SUPERCEDE_FLAG: If set and an associate request comes in while there is already an association, directs LEC Configuration Manager to abort the old association and accept the new one. If not set, LEC Configuration Manager refuses the new association.</p> <p>ALLOW_MULTI_FLAG: If set and an associate request comes in while there is already an association, directs LEC Server to keep the old association and also accept the new one. If not set, refuse the new association and keep the old one.</p>	
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 Template)

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 LEC Server. This name will be displayed throughout LEC	Yes

Parameter	Definition	Required
	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 LEC Configuration Manager, for example, when connecting this node to another node.	
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 Template)

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	Yes

Parameter	Definition	Required
	<p>data within LEC Server. This name will be displayed throughout LEC 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 LEC Configuration Manager, for example, when connecting this node to another node.</p>	
Remote VCC	<p>Specifies the common name of the remote VCC.</p>	Yes
Pclass	<p>Specifies the LiveData poll class controlling the frequency of issuing the identify request.</p> <ul style="list-style-type: none"> ▪ 1 - specifies 10 seconds. ▪ 2 - specifies 1 seconds. ▪ 3 - specifies .1 seconds. <p>Do not use poll class 0 because it is the PUSH class, which makes no sense in this case.</p>	Yes
Timeout	<p>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.</p> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p>Note: Because this parameter is global, specify this parameter only once</p> </div>	No